PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-257048
(43)Date of publication of application : 25.09.1998

(51)Int.Cl. H04L 9/32
G06F 15/00

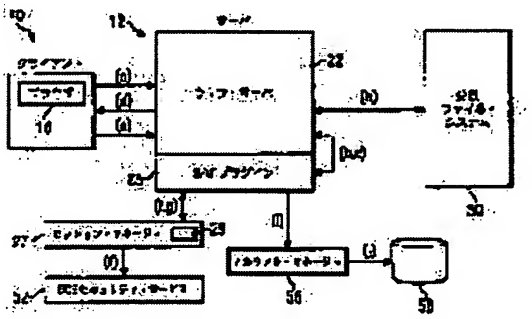
(21)Application number : 10-004566
(22)Date of filing : 13.01.1998

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>
(72)Inventor : SHAU-BEN SHI
MICHAEL BRADFORD OLT
ARNST ROBERT PLASMAN
BRUCE ARLAND RICH
MCKEELER AN ROSIRES
THEODORE JACQUES LONDON SHELLERD

(30)Priority
Priority number : 97 790041 Priority date : 28.01.1997 Priority country : US

(54) AUTHENTICATION FOR DISTRIBUTED FILE SYSTEM WEB SERVER USER BY COOKIE

(57)Abstract:
PROBLEM TO BE SOLVED: To authenticate a user for accessing a distributed file system through a web server by storing a qualification proof in a data base and returning a connection client state object provided with an intrinsic identifier to a web client in the case that the user can be authenticated by the web server.
SOLUTION: In the case that a request is from a browser 16 which supports a cookie, a web server platform 12 judges whether or not the existing cookie is provided inside a request header, and in the case of not being provided, urges a user ID and a password to the user and returns the cookie provided with the URL of a document requested by the user as an entry. The user is authenticated by using the user ID and the password and the qualification proof used for acquiring access to the distributed file system 50 by the user is generated. Thus, a web document inside the system 50 is retrieved by using the qualification proof.



LEGAL STATUS

[Date of request for examination] 16.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]	3003997
[Date of registration]	19.11.1999
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a client and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested a) Reception of the user ID from said client by said web server and a password is answered. The step which memorizes the rating certification which performs a log in protocol with said security service, and is produced as a result, b) The step which returns the self-sustaining client condition object which has an identifier in said client, c) Approach containing the step which gains continuing access to the web document in said distributed file system when said client uses said self-sustaining client condition object containing said identifier instead of said user ID and a password.
[Claim 2] The approach according to claim 1 used in order that said identifier in said self-sustaining client condition object may search the rating certification memorized at said step to memorize.
[Claim 3] The approach according to claim 1 by which said web server is provided with said user ID and password by HTML format.
[Claim 4] The approach according to claim 3 by which said HTML format is completed by the user of said client.
[Claim 5] To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a web client, and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested a) The HTTP demand received by said web server is answered. The step which judges whether said web client has the browser which supports a self-sustaining client condition object, b) When said web client has the browser which supports said self-sustaining client condition object, The step at which the 1st self-sustaining client condition object containing URL from which said web server is discriminated by said web client by log in HTML format and said HTTP demand is transmitted, c) The step to which said user completes said HTML format with user ID and a password, d) The step which returns the completed format to said web server together with said 1st self-sustaining client condition object containing said URL, e) Extract information from said completed format and a log in protocol is performed with said security service. The step which generates rating certification, and the step which returns the 2nd self-sustaining client condition object which has an identifier in the f aforementioned web client, g) Approach containing the step which gains continuing access to the web document in said distributed file system when said web client uses said 2nd self-sustaining client condition object containing said identifier instead of user ID and a password.
[Claim 6] The approach according to claim 5 used in order that said identifier may access said rating certification.
[Claim 7] To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a web client, and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested a) Reception of the transaction request from said web client is answered. The step which judges whether a log in protocol is performed with said security service, and said web client has an access privilege to said distributed file system, b) The step which returns an error message to said web client when said web client does not have an access privilege to said distributed file system, c) When said web client has an access privilege to said distributed file system, The step which memorizes the rating certification generated as a result of said log in protocol in the database of the rating certification related with an attested user, d) The step which returns the Cookie which has the identifier related with said web client by said web client at a proper, e) Approach containing the step from which said client gains continuing access to the web document in said distributed file system by using said Cookie instead of user ID and a password.
[Claim 8] The approach according to claim 7 used in order that said identifier in said Cookie may search the rating certification memorized from said database at said step to memorize.
[Claim 9] To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a web client, and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested The step which holds to storage said rating certification of the user who had access to said distributed file system attested, Reception of the self-sustaining client condition object which has an identifier from said web client is answered. How to contain the step which accesses one of the rating certification in said storage, and the step which accesses the file in said distributed file system using said rating certification using said identifier.
[Claim 10] To a web server connectable with the distributed file system of a distributed computer environment It is the computer program product used in order to attest a web client. In that in which said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested A computer read-out possible storage and the program data memorized by said computer read-out possible storage are included. Said program data answer reception of the user ID from said web client by said web server, and a password. A means to memorize the rating certification which performs a log in protocol with said security service, and is produced as a result, A means to return the self-sustaining client condition object which has an identifier in said web client, The computer program product which answers reception of said self-sustaining client condition object containing said identifier, and includes a means to control continuing access to the web document in said distributed file system.
[Claim 11] A computer program product including a means by which said program data answer said log in protocol, and generate an error message according to claim 10.
[Claim 12] The computer program product according to claim 10 with which said program data include a means to establish storage of said rating certification of the user attested by said distributed file system.
[Claim 13] The computer program product according to claim 10 said whose self-sustaining client condition object is Cookie.
[Claim 14] To a web server connectable with the distributed file system of a distributed computer environment It is the computer program product used in order to attest a web client. In that in which said distributed computer environment contains the security

service which returns rating certification to the user who had access to said distributed file system attested A computer read-out possible storage and the program data memorized by said computer read-out possible storage are included. A means by which said program data hold storage of said rating certification of the user who had access to said distributed file system attested, By answering reception of the self-sustaining client condition object which has an identifier from said web client, and accessing one of said the rating certification in said storage using said identifier A computer program product including the means which enables access of the web document in said distributed file system.

[Claim 15] It is a computer connectable with the distributed computer environment containing the security service for returning rating certification to the user who had access to a distributed file system attested. A processor, An operating system and a stateless computer network are minded. The web server program which provides a web client connectable with a web server program with World-Wide-Web information retrieval, Server plug-in which attests said web client to said web server program is included. Said server plug-in answers reception of the user ID from said web client by said web server, and a password. A means to memorize the rating certification which performs a log in protocol with said security service, and is produced as a result, A means to return the self-sustaining client condition object which has an identifier in said web client, The computer which answers the reception which said self-sustaining client condition object containing said identifier follows instead of user ID and a password, and includes a means to control access to the web document in said distributed file system.

[Claim 16] The computer according to claim 15 by which said means to control accesses said rating certification using said identifier.

[Claim 17] It is the approach of accessing a document from the distributed file system to which a web server and said web server are connected. In what has the security service with which said distributed file system supported within a distributed computer environment returns rating certification to the user who had access to said distributed file system attested a) Reception of the user ID from said web client by said web server and a password is answered. The step which memorizes the rating certification which performs a log in protocol with said security service, and is produced as a result, b) The step which returns the self-sustaining client condition object which has an identifier in said web client, c) when said client uses said self-sustaining client condition object containing said identifier instead of said user ID and a password The approach containing the step which gains access to the web document in said distributed file system, and the step from which the d aforementioned web client gains access to the web document in said web server using said user ID and password.

[Claim 18] The approach containing the step holding storage of said rating certification of the user who had use of said distributed file system attested according to claim 17.

[Claim 19] The approach according to claim 18 used in order that said identifier may search said rating certification from said storage.

[Claim 20] The approach according to claim 17 of containing the step which provides said web client with the error message of a special order from said web server, when said log in protocol is unsuccessful.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] It is related with generally this invention enabling access to the web document especially memorized by the safe distributed file system about web transaction processing.
[0002]
[Description of the Prior Art] The World Wide Web of the Internet is the most successful distributed application in the history of a computer. In a web environment, a client machine uses a hyper-text transfer protocol (HTTP) for the transaction to a web server, and this is a known application protocol which offers the user access to files (for example, a text, graphics, an image, a sound, video, etc.) using the Page Description Language of the criterion known as a hyper-text markup language (HTML). HTML offers the formatting of original source documents and a developer is enabled to specify the "link" to other servers and files. The network path to a server is identified in the paradigm of the Internet by the so-called uniform resource locator (URL) which has the special functor of defining network connection. Use of the browser (for example, Netscape Navigator) compatible with HTML in a client machine includes assignment of the link through URL. According to it, a client gives a demand to the server identified in a link, and receives the document formatted into collateral according to HTML.
[0003] Many organizations use two or more computers which interconnect within a distributed computer environment, and a user accesses a distributed resource and processes application there. The known distributed computer environment called DCE has been realized using available software from OSF (Open Systems Foundation). Much application may be used in order to offer distributed service of data common use, printing service, database access, etc., if the DCE environment comes to be chosen as solution in a company. OSF DCE contains the distributed file system called the distributed file service (DFS) used in these environments.
[0004] DFS offers many advantages in which the file server of a standalone version is excelled. The protection of information by data and the high availability of a resource, the capacity that shares information over the whole super-large system, and the steadfast DCE security device etc. is contained in them. Especially, DFS makes a file usable to altitude through a duplicate, and even if one of the machines by which a file is arranged breaks down, it makes it possible to access the copy of a file. DFS gathers up all the files memorized by various file systems again in global-area name space. Two or more servers can export those file systems to this name space. All DFS users share this name space soon, and all DFS files become usable easily from the DFS client machine of arbitration.
[0005] In order to use the scalability, file availability, and security device of DFS (or other similar distributed file systems), it is very desirable to extend the function of the web server of the existing standalone version in a company environment. The user who has an off-the-shelf browser (namely, ready-made) as a by-product can access easily without the software of the addition on a client machine the web information memorized in DFS name space. However, in order to attain this target, it is required to unify with the conventional DFS security the security device offered by the web server. One of the options is using a basic (provided by web server) authentication (Basic Authentication) device, and gaining user ID and a password to each HTTP demand. However, there are some problems in using a known basic authentication device in the situation of DFS.
[0006]
[Problem(s) to be Solved by the Invention] Especially, user ID and a password are passed to all demands. Therefore, they tend to be attacked by the invader even if a password is protected according to a specific encryption device (for example, SSL). It is difficult 2nd for DFS and a web server security device to live together. A browser memorizes the user ID and the password which are transmitted to a specific server, and user ID and a password are added to all HTTP demands transmitted to the server. When the device in which a distributed file system is made to access a web server is offered, a web server will hold both the document (protected by web server security) memorized on a server local directory, and the document (protected by DFS security) memorized on DFS. If it sees from a browser, a web server will be a single server and will only memorize one pair of user ID and the password to the web server. When the user is browsing both the DFS document and the web server document, a user will be reminded of user ID and a password, the change in a web server document from a DFS document, and whenever [its] it is reverse. Finally, when DFS authentication goes wrong, only the restricted error information may be returned to a user.
[0007] These problems make a known basic authentication device unsuitable, when unifying a web server and a DFS security device. This invention solves this problem.
[0008] The 1st purpose of this invention is attesting the user who accesses a distributed file system through the Internet World-Wide-Web server.
[0009] Another purpose of this invention is offering the distributed file system authentication device for web browsing only a transfer of user ID and a password being required, when a user logs in to a file system for the first time through a web server. To the continuing demand, the secret handle memorized by "Cookie (cookie)" is transmitted to a web server from a web browser.
[0010] Furthermore, another purpose of this invention is making easy safe web document access from a distributed file system by using a self-sustaining client condition HTTP Cookie authentication device.
[0011] Furthermore, if a user is already log in ending at DFS when a user changes from a DFS document to a web server document, another purpose of this invention is realizing the authentication device of the Cookie base for the DFS web server application which coexists with a known basic authentication security device, as it does not press for user ID and a password.
[0012] Furthermore, another purpose of this invention is offering the customized error message which is transmitted to a browser from a web server instead of the error message offered according to a known basic authentication device.
[0013] The more general purpose of this invention is unifying with the conventional DFS security the security device offered by the web server. In a company environment, this raises the function of the web server of the existing standalone version so that the scalability, file availability, and security device of DFS (or other similar distributed file systems) may be used. The user who has an

off-the-shelf browser as a by-product can access easily without the software of the addition on a client machine the web information memorized in DFS name space.
[0014]
[Means for Solving the Problem] It is provided by these approaches of this invention that reach and other purposes attest a web client to a web server connectable with the distributed file system of a distributed computer environment. A distributed computer environment contains the security service which returns rating certification to the user who had access to a distributed file system attested. Reception of the user ID from the web client by the web server and a password is answered, and a log in protocol is performed with security service. When a user may be attested, the Inn memory rating certification database of rating certification with which rating certification is related with an attested user memorizes. Next, a web server returns the self-sustaining client condition object which has the identifier of a proper in a web client. Although this object is referred to as Cookie by the way, it is used in order for this to enable a web client next to browse the web document in a distributed file system. When you wish to publish the demand which a web client follows especially to a distributed file system, the self-sustaining client condition object containing an identifier is used instead of user ID and a password, and this makes a session much more safe. In this actuation, a Cookie identifier is used as a pointer to the Inn memory rating certification database, and in order that rating certification may be searched next and may make easy multiple-files access from a distributed file system, it is used.
[0015] A web client can still gain access to a web server (distributed file system document by contrast) document through the conventional user ID and the conventional password in a HTTP demand to coincidence.
[0016] According to the suitable approach of this invention, it is judged first whether an initial HTTP demand is answered and it has the browser to which a web client supports a self-sustaining client condition object, i.e., "Cookie." When it has, a web server transmits log in HTML format and the 1st Cookie containing URL identified by HTTP demand to a web client. Next, with his user ID and password, it presses for a user so that HTML format may be completed. Then, a web client returns the completed format to a web server together with the 1st Cookie. In a web server, information is extracted from the completed format and the log in protocol of a distributed file system is supplied. If a log in is successful, user rating certification will be generated and the Inn memory rating certification database will memorize suitably. When a log in is unsuccessful, an error message is returned to a web client. Next, a proper identifier is generated to an attested user and this is used as a pointer to a rating certification database. Next, this identifier is arranged in the new Cookie transmitted to a web client. New Cookie is used next by the web client for all continuing file accesses to a distributed file system. By using new Cookie, a web client does not need to transmit repeat user ID and a password through a network. However, a client can still use user ID and a password, and can gain simple file access from a web server (a distributed file system by contrast).
[0017] The above-mentioned explanation describes some outlines of the purpose and the description that this invention is related. These purposes may be gained like the after-mentioned that the useful result of many others applies this invention indicated to another appearance, or by changing this invention by expressing some of descriptions which were only excellent in this invention, and applications. Therefore, other purposes of this invention and a perfect understanding will be obtained by referring to the gestalt of operation of below-mentioned this invention.
[0018]
[Embodiment of the Invention] The typical system by which this invention is realized is shown in drawing 1 . A client machine 10 is connected to the web server platform 12 through a communication channel 14. A communication channel 14 is the Internet, intranet, or other known connection on account of explanation. In the case of the Internet, the web server platform 12 is one of two or more of the accessible servers by the client, and one of the clients is shown by the machine 10. A client machine is a known software tool used in order that this may access a network server including a browser 16. As an example, a client machine is a personal computer. A network SUKEBU navigator (all versions), Microsoft Internet Explorer (all versions), etc. are contained in a typical browser, and these each is software programs in which off-the-shelf ["off-the-shelf"] or download is possible. A web server platform (sometimes referred to as a "web" site) supports a file with the format of a hyper-text document and an object. The network path to a server is identified by the so-called uniform resource locator (URL) in the paradigm of the Internet. World Wide Web is the multimedia information retrieval system of the Internet. Especially this is the set of the server of the Internet which uses a hyper-text transfer protocol (HTTP), and HTTP offers the user access to a file using a hyper-text markup language (HTML).
[0019] The typical web server platform 12 is IBM. RISC This executes the web server programs 22, such as the AIX (4.1 or above extended dialogue executive versions) operating system 20 and the Netscape ENTAPU rise version 2.0 which supports an interface escape, including System/6000 computer 18. The web server platform 12 includes the graphical user interface (GUI) 24 for management further. Various models of the computer of the RISC base are described by many publications of IBM, for example, "RISC System/6000, 013 and 7016 POWERstation and POWERserver Hardware Technical Reference", and the order number SA 23-2644-00. AIX OS is described by "AIX Operating System TechnicalReference" (the 1st edition, November, 1985) of the IBM issue etc. Although the above-mentioned platform is useful, the combination of other suitable hardware / operating system / web servers of arbitration may be used.
[0020] A web server accepts a client demand and returns a response. Actuation of a web server 18 is managed by much server application functions (SAF), and each SAF is constituted so that it may perform in the specific step of a sequence. This sequence is shown in drawing 2 , it starts at the authorization conversion (authorization translation) step 30, and the authorization information on arbitration that a server is transmitted by the client between them is changed into a user and a group. If needed, an authorization conversion step decodes a message and gains an actual client demand. It is called name conversion, and URL related with a demand is maintained as it is, or step 32 may be changed into the file name of system dependence, Redirection URL, or a mirror site URL. Step 34 is called a path check, a server performs various tests for the path of a result, and it is guaranteed that a given client can search a document. By the way, step 36 is referred to as an object type, and the MIME (multiple-purpose Internet mail escape) type information (for example, a text/html, an image/gif, etc.) over a given document is identified. Step 38 is called service, and a web server routine chooses internal server ability, and returns a result to a client through the server service routine of normal. It depends on the property of a demand for the specific function chosen. Step 40 is called a log addition and the information about a transaction is recorded. When step 42 is called an error and an error is encountered, a server answers a client. Detail of these actuation is given by "Web Server Programmer's Guide" of the Netscape company issue, and Chapter 5.
[0021] A web server 18 includes the known set of a server application function (SAF). These functions return the demand of a client, and other configuration data of a server to reception, and return a response to a server as an output as an input. if drawing 1 is referred to again, a web server 18 will offer the escape to which this is enabled for an application developer to lead the software program generally referred to as "plug-in", and to extend and (or) customize a core function (namely, SAF) including the application programming interface (API) 26. This invention uses a server API 26, plug-in which makes a user's authentication easy is offered, and, thereby, as for the user of a client machine 10, web access to the document on a distributed file system 50 is attained using a browser.

[0022] According to the general purpose of especially this invention, the user of a client machine 10 uses the off-the-shelf (or unconsciously [Intention target]) browser 16, and makes it possible to access, browse and search the document arranged in a distributed file system 50. Such one file system 50 is distributed file service (DFS), and this is a known distributed file system realized in the network environment called a distributed computer environment (DCE). DCE is realized using available software from OSF. In a distributed computer environment, the group of a machine is usually referred to as a "domain." OSF A DCE domain is called a "cel." A DCE cel may be a complicated environment containing hundreds of machines which exist in much location.
[0023] DCE DFS50 -- a naming sake -- a remote procedure call (RPC) -- moreover, data utility service is offered by using the DCE security service 52 for authentication service. DFS50 carries out an interface to the DCE security service 52 through the session manager process 27. This is explained in full detail by the United States patent application 08th / No. 790042. In addition to use of DCE service, the own device of DFS is abundant. By making it possible to view the file space where DFS offers uniform global-area file space, and all DFS client users are the same as for this, and carrying out the cache of the file system data in a client, the network traffic to a file server is reduced and scalability and the engine performance are improved. DFS supports one of the devices in the capacity which exports a notice file locking and the native file system of an operating system again. For example, in the case of an AIX operating system, a native file system is a jar NARUDO file system (JFS). Furthermore, DFS offers, the physical file system (LFS), i.e., the DCE local file system, of itself. DCE LFS provides the support of the DCE access control list (ACL) about the file and directory for protecting access to data, and a list with advanced data control capacity, such as a duplicate and load equilibration.
[0024] DFS50 uses the so-called authentication of the DCE Kerberos base (Kerberos-based). The "rating certification" of UNIX is related with each file manipulation, and holds the local authentication information on the actuation. Especially rating certification is DS which defines a specific machine (or user on a multiuser machine). From a viewpoint of a local operating system, rating certification contains arbitrarily user ID, group ID, and the list of privileges of an operating system and the authentication identifier known as PAG (process authentication group). PAG functions between DFS50 and the DCE security service 52 as a tag which associates a "ticket." When a DFS user attests through a known DCE log in device as dce_login, DCE security service has a dialog through DFS (minding a network) and a setpag() interface, and PAG / ticket relation to the rating certification of a process are established. On the occasion of a file system demand, DFS extracts PAG from rating certification structure, and establishes a DCE user's authentication to the RPC demand to a DFS file server.
[0025] The flows of control related with this invention are shown in the process flow Fig. of drawing 3 R> 3. This drawing shows the basic system of drawing 1 , and contains the account manager 56 who has the related database 58. It starts at the time of initialization of a web server, and the session manager 27 is suitably performed by workstation computer 18. The session manager 27 includes the storage area 29 of itself. When a client 10 requires a DFS (leading browser 16) document (step a), a web server 22 calls a server (using SAF plug-in 25) path check (step b). It judges whether a path check has DCE rating certification with a suitable user by the session manager 27. In negation (step c), the SAF plug-in 25 returns an error message (for example, "401; unauthorized") to a browser 16 (step d), and reminds a user of user ID and a password. After gaining user ID and a password from a user (step e), the SAF plug-in 25 calls the session manager 27 (step f), and gains a user's DCE rating certification. The session manager 27 returns DCE rating certification to a web server 22 (step g). Next a server uses this user rating certification showing a user, and searches the document memorized by DFS50 (step h). The account (using another API plug-in suitably) manager 56 is called after searching a document (step i), and suitable use information is kept in a database 58 (step j).
[0026] When trying in order that a user may access a DFS file, the session manager 27 is called by the web server. When the user is already attested by DCE, the session manager 27 returns user rating certification to a server, and a server uses this rating certification and searches a DFS document for a user. When the user is not attested, the session manager 27 logs in for a user, and gains rating certification from DCE security. A session manager holds an in-memory database, the user who logged in is pursued, and, thereby, a user can access two or more DFS pages.
[0027] Instead of using a basic authentication device, this invention uses self-sustaining client condition HTTP Cookie. Cookie is the known Internet device which can use the connection by the side of a server (CGI script etc.), in order to memorize and retrieve the information on a client side. A server can also transmit status information again, when returning a HTTP object to a client. A client memorizes this status information. Usually, the condition object called "Cookie" may include description of the range of URL with the effective condition. According to "Persistent Client State HTTP Cookies" and Preliminary Specification which are seen by pass"/newref/std/cookie_spec_html" of netscape.com, Cookie is usually introduced into a client by including a Set_Cookie header as a part of HTTP response through a CGI script. Known Cookie functor is shown below.
[0028] Functor of a Set-Cookie HTTP response header: This is the format of a CGI script for adding the new data memorized by the client for next retrieval to a HTTP header.
Set-Cookie:NAME=VALUE;expires=DATE;
path=PATH;domain=DOMAIN_NAME;secure [0029] NAME=VALUE -- this string is a semicolon, a comma, and an alphabetic character sequence except a null. When such data need to be arranged in a name or a value, the specific coding approaches, such as URLstyle%XX coding, are recommended. However, coding is not defined or required. This is the only attribute demanded on a Set_Cookie header.
[0030] An expires=DATEexpires (term) attribute specifies the data string who defines the service life of the Cookie. If it reaches on the expiration date, Cookie will not be memorized or distributed any longer. The date string is as follows.
Wdy, DD-Mon-YYY HH:MM:SS GMT [0031] domain=DOMAIN_NAME -- when looking for Cookie Liszt in quest of effective Cookie, the domain (domain) attribute of Cookie is compared with the Internet domain name of the host to whom the fetch of the URL is carried out. If a tail is in agreement, Cookie will check whether it should be transmitted or not through path matching. "Tail matching" means that a domain attribute is matched to a host's tail of a completely proper domain name. For example, the domain attribute of "acme.com" is matched with host names, such as "shipping.crate.acme.com" and "anvil.acme.com."
[0032] Only the host in the domain specified can set Cookie to a domain, and a domain must have at least two or three periods, in order to avoid the domain of the format of ".com", ".edu", and "va.us." The domain of the arbitration included in one of the seven special top level domains shown below needs only two periods. all -- others -- a domain needs at least three periods. Seven special top level domains are "COM", "EDU", "NET", "ORG", "GOV", "MIL", and "INT." The default of a domain is the host name of the server which generated the Cookie response.
[0033] A path=PATHpath (path) attribute is used in order to specify the subset of URL in the domain where Cookie is effective. When Cookie is already in agreement by domain matching, the path name element of URL is compared with a path attribute, and when coincidence exists, it is considered that Cookie is effective and it is transmitted together with a URL demand. Path"/foo" is in agreement with "/foobar" and "/foo/bar.html." Path"/" is the most general path. When a path is not specified, it is assumed that it is the same path as the document described by the header containing Cookie.
[0034] the case where secure Cookie is marked with secure (insurance) -- case a communication channel with a host is safe for this --

as long as -- it is transmitted. Current and this mean that safe Cookie is transmitted only to a HTTPS (HTTP through SSL) server. When secure is not specified, even if a non-protecting channel top is transmitted to Cookie by the plaintext, it considers that it is safe.
[0035] Functor of a Cookie HTTP request header: When requiring URL from a HTTP server, and a browser matches URL to all Cookie and its either of those corresponds, Rhine containing the pair of the name/value of all the congruous Cookie is included in a HTTP demand. The format of the Rhine is shown below.
Cookie:NAME1=OPAQUE_STRING1;NAME2=OPAQUE_STRING2 [0036] The flow chart which shows the authentication flow of this invention using HTTP Cookie is shown in drawing 4 . A routine is started at step 60 to each HTTP demand received by the server. It is judged whether it was transmitted at step 62 by the browser to which a demand supports HTTP Cookie. For example, although both Netscape browser (for example, navigator (all versions)) and Microsoft browser (for example, Microsoft Internet Explorer (all versions)) support Cookie, the browser program of other marketing is not supported. When the result of the test of step 62 is negation, it is used in order that basic authentication may attest a user at step 64. When the test result of step 62 is affirmation (that is, a browser supports Cookie), this approach is continued to step 66 and it tests whether the existing Cookie is contained in a request header. When the test result of step 66 is affirmation, the user is already attested and basic authentication is used. Although a browser supports Cookie when the result of step 66 is negation, Cookie does not exist yet.
[0037] At step 68, a server returns log in HTML format and reminds a user of user ID and a password. A server returns the Cookie which contains as an entry URL of the document demanded by the user again. After user ID is especially attested by the DCE security server as mentioned above (minding a session manager), a web server needs to search a document for a user. In this case, a web server needs original URL, in order to search a document. Since a web server is statelessness, a browser must be provided with original URL. This is attained by offering Cookie. At step 70, a user enters user ID and a password in HTML format. The format itself is generated like known using a CGI script. At step 72, the user ID and the password which were offered in format are returned to a server together with the Cookie which the browser received at step 68.
[0038] Using user ID and a password, it continues to step 74 and a routine attests a user through the conventional dce_login device. Like known, activation of dce_login generates the "rating certification" used in order that a user may gain access to DFS. When un-succeeding [of authentication] is judged as a result of step 76, a server is step 80 and returns the customized HTML document which describes a specific failure to a browser. Next, the Cookie generated at step 68 is destroyed at step 81. When a success of authentication is judged at step 76, it continues to step 77 and a routine generates ID (for example, DCE UUID) of a user's proper. At step 78, the DFS rating certification generated by the log in (to DCE security server) is memorized by the database (suitably in-memory storage) related with a session manager, and an indexing is carried out by ID of a proper.
[0039] It continues to step 82 and a routine returns the new Cookie containing ID of the proper generated by the browser at step 77. Next, the Cookie generated at step 68 is destroyed at step 83. ID of a proper is a secret handle in fact, and this is an entry to the table of the rating certification memorized by the database related with a session manager. To the continuing demand to the service from a browser, ID (supported within the new Cookie returned to the browser from the server at step 82) of a proper is used as a pointer indicating a user's DFS rating certification memorized by this database. Therefore, a server receives the new demand which has the new Cookie containing ID of a proper at step 84. At step 86, in order that ID of this proper may gain a user's rating certification, it is used. It is used in order to search with step 88 the web document with which rating certification is supported within DFS (web server suitably under pretense of a browser).
[0040] The demand which continues from a browser transmits the Cookie which has ID of a proper, therefore steps 84, 86, and 88 are repeated to all continuing demands. Therefore, according to this invention, it is required, when a transfer of user ID and a password logs in first only once and a user logs in to DFS. Then, it is transmitted on the occasion of the demand which the Cookie which has ID of a proper follows. This device can coexist with the basic authentication security device offered by the web server. If the user has already logged in through DCE security service when changing from a DFS document to a web server document, he will not be again reminded of user ID and a password. The customized error message may be returned to a browser, without being restricted to the error code specified in a basic authentication device.
[0041] One of the suitable examples of the authentication device of the Cookie base of this invention exists in the random access memory of a computer as an instruction set in a code module (program code). An instruction set may be memorized in [, such as an optical disk (used by CD ROM drive) or a floppy disk (used by the floppy disk drive), / which can be removed] another computer memory, for example, a hard disk drive, or memory, or may be downloaded through a computer network until it is required by computer. Furthermore, although various approaches mentioned above are alternatively realized conveniently by software in activation or the general purpose computer reconfigurated, it will be understood by this contractor that such an approach may be realized by hardware, firmware, or the special equipment constituted so that the approach step demanded might be performed.
[0042] It connects with computer networks, such as the Internet, directly or indirectly, or the "web" client should be widely interpreted as meaning the computer of the known of arbitration, or arbitration connectable in the format developed behind, or its component so that it may be used on these specifications. The vocabulary "web" server should also be widely interpreted so that a computer, a computer platform, a computer, the add-on of a platform, or the component of the arbitration may be meant.
[0043] Furthermore, although this invention has been described about the suitable example in a specific distributed file system environment, when this invention follows modification on the meaning and within the limits, it will be understood by this contractor that it may realize also in other different hardware and operating system architecture. It follows, for example, although the off-the-shelf browser was realized so that this invention might be suitably accessible in the web document memorized by DFS, the principle of this invention is applicable like other known architecture, such as AFS (DFS was drawn), not to mention the Network File System (NFS) developed by Sun Microsystems, Inc. Furthermore, OSF DCE is not the requirement of this invention, either.
[0044] As a conclusion, the following matters are indicated about the configuration of this invention.
[0045] (1) To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a client and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested a) Reception of the user ID from said client by said web server and a password is answered. The step which memorizes the rating certification which performs a log in protocol with said security service, and is produced as a result, b) The step which returns the self-sustaining client condition object which has an identifier in said client, c) Approach containing the step which gains continuing access to the web document in said distributed file system when said client uses said self-sustaining client condition object containing said identifier instead of said user ID and a password.
(2) The approach of the aforementioned (1) publication used in order that said identifier in said self-sustaining client condition object may search the rating certification memorized at said step to memorize.
(3) The approach of the aforementioned (1) publication that said web server is provided with said user ID and password by HTML format.
(4) The approach of the aforementioned (3) publication that said HTML format is completed by the user of said client.

(5) To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a web client, and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested a) The HTTP demand received by said web server is answered. The step which judges whether said web client has the browser which supports a self-sustaining client condition object, b) When said web client has the browser which supports said self-sustaining client condition object, The step at which the 1st self-sustaining client condition object containing URL from which said web server is discriminated by said web client by log in HTML format and said HTTP demand is transmitted, c) The step to which said user completes said HTML format with user ID and a password, d) The step which returns the completed format to said web server together with said 1st self-sustaining client condition object containing said URL, e) Extract information from said completed format and a log in protocol is performed with said security service. The step which generates rating certification, and the step which returns the 2nd self-sustaining client condition object which has an identifier in the f aforementioned web client, g) Approach containing the step which gains continuing access to the web document in said distributed file system when said web client uses said 2nd self-sustaining client condition object containing said identifier instead of user ID and a password.
(6) The approach of the aforementioned (5) publication used in order that said identifier may access said rating certification.
(7) To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a web client, and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested a) Reception of the transaction request from said web client is answered. The step which judges whether a log in protocol is performed with said security service, and said web client has an access privilege to said distributed file system, b) The step which returns an error message to said web client when said web client does not have an access privilege to said distributed file system, c) When said web client has an access privilege to said distributed file system, The step which memorizes the rating certification generated as a result of said log in protocol in the database of the rating certification related with an attested user, d) The step which returns the Cookie which has the identifier related with said web client by said web client at a proper, e) Approach containing the step from which said client gains continuing access to the web document in said distributed file system by using said Cookie instead of user ID and a password.
(8) The approach of the aforementioned (7) publication used in order that said identifier in said Cookie may search the rating certification memorized from said database at said step to memorize.
(9) To a web server connectable with the distributed file system of a distributed computer environment In that in which it is the approach of attesting a web client, and said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested The step which holds to storage said rating certification of the user who had access to said distributed file system attested, Reception of the self-sustaining client condition object which has an identifier from said web client is answered. How to contain the step which accesses one of the rating certification in said storage, and the step which accesses the file in said distributed file system using said rating certification using said identifier.
(10) To a web server connectable with the distributed file system of a distributed computer environment It is the computer program product used in order to attest a web client. In that in which said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested A computer read-out possible storage and the program data memorized by said computer read-out possible storage are included. Said program data answer reception of the user ID from said web client by said web server, and a password. A means to memorize the rating certification which performs a log in protocol with said security service, and is produced as a result, A means to return the self-sustaining client condition object which has an identifier in said web client, The computer program product which answers reception of said self-sustaining client condition object containing said identifier, and includes a means to control continuing access to the web document in said distributed file system.
(11) The computer program product of the aforementioned (10) publication including a means by which said program data answer said log in protocol, and generate an error message.
(12) The computer program product of the aforementioned (10) publication with which said program data include a means to establish storage of said rating certification of the user attested by said distributed file system.
(13) The computer program product of the aforementioned (10) publication said whose self-sustaining client condition object is Cookie.
(14) To a web server connectable with the distributed file system of a distributed computer environment It is the computer program product used in order to attest a web client. In that in which said distributed computer environment contains the security service which returns rating certification to the user who had access to said distributed file system attested A computer read-out possible storage and the program data memorized by said computer read-out possible storage are included. A means by which said program data hold storage of said rating certification of the user who had access to said distributed file system attested, By answering reception of the self-sustaining client condition object which has an identifier from said web client, and accessing one of said the rating certification in said storage using said identifier A computer program product including the means which enables access of the web document in said distributed file system.
It is a computer connectable with the distributed computer environment containing the security service for returning rating certification to the user who had access to a distributed file system attested. (15) A processor, An operating system and a stateless computer network are minded. The web server program which provides a web client connectable with a web server program with World-Wide-Web information retrieval, Server plug-in which attests said web client to said web server program is included. Said server plug-in answers reception of the user ID from said web client by said web server, and a password. A means to memorize the rating certification which performs a log in protocol with said security service, and is produced as a result, A means to return the self-sustaining client condition object which has an identifier in said web client, The computer which answers the reception which said self-sustaining client condition object containing said identifier follows instead of user ID and a password, and includes a means to control access to the web document in said distributed file system.
(16) The computer of the aforementioned (15) publication by which said means to control accesses said rating certification using said identifier.
(17) It is the approach of accessing a document from the distributed file system to which a web server and said web server are connected. In what has the security service with which said distributed file system supported within a distributed computer environment returns rating certification to the user who had access to said distributed file system attested a) Reception of the user ID from said web client by said web server and a password is answered. The step which memorizes the rating certification which performs a log in protocol with said security service, and is produced as a result, b) The step which returns the self-sustaining client condition object which has an identifier in said web client, c) when said client uses said self-sustaining client condition object containing said identifier instead of said user ID and a password The approach containing the step which gains access to the web document in said distributed file system, and the step from which the d aforementioned web client gains access to the web document in

said web server using said user ID and password.
(18) The approach of the aforementioned (17) publication containing the step holding storage of said rating certification of the user who had use of said distributed file system attested.
(19) The approach of the aforementioned (18) publication used in order that said identifier may search said rating certification from said storage.
(20) The approach of the aforementioned (17) publication which contains the step which provides said web client with the error message of a special order from said web server when said log in protocol is unsuccessful.

[Translation done.]

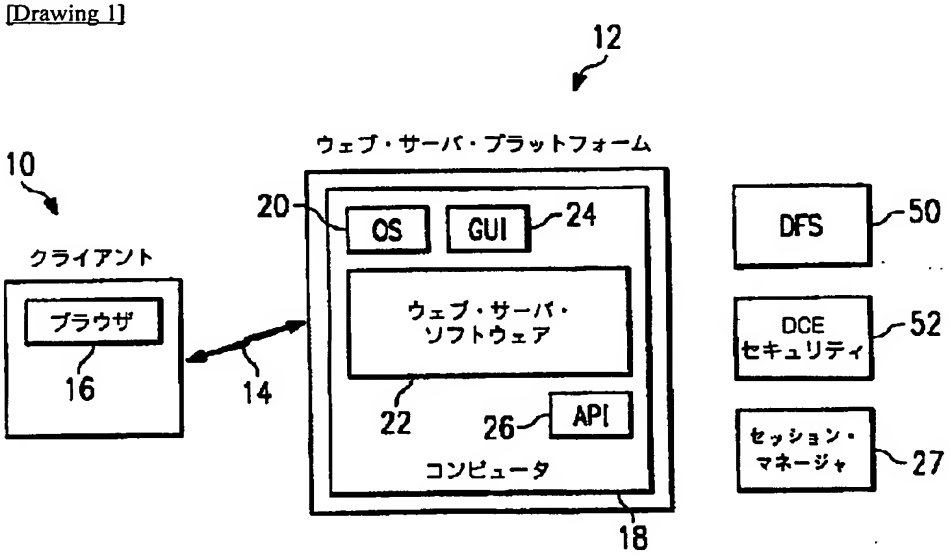
DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is drawing showing the typical system by which plug-in of this invention is realized.
[Drawing 2] It is drawing which answers reception of the demand from the browser of a client machine and in which showing the flow chart of actuation by the side of the server of the conventional web transaction.
[Drawing 3] It is the process flow Fig. showing the web transaction realized according to instruction of this invention.
[Drawing 4] It is drawing showing the detail flowchart of the process flow of this invention.
[Description of Notations]
10 Client Machine
12 Web Server Platform
14 Communication Channel
16 Browser
18 Computer
20 AIX Operating System
22 Web Server Program
24 Graphical User Interface (GUI)
25 SAF Plug-in
26 Application Programming Interface (API)
27 Session Manager
29 Storage Area
50 Distributed File System
52 DCE Security Service
56 Account Manager
58 Database

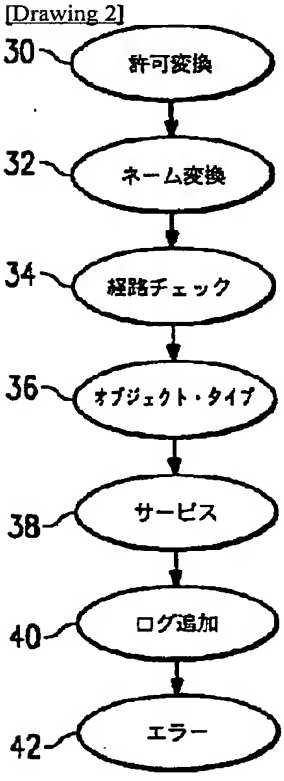
[Translation done.]

DRAWINGS

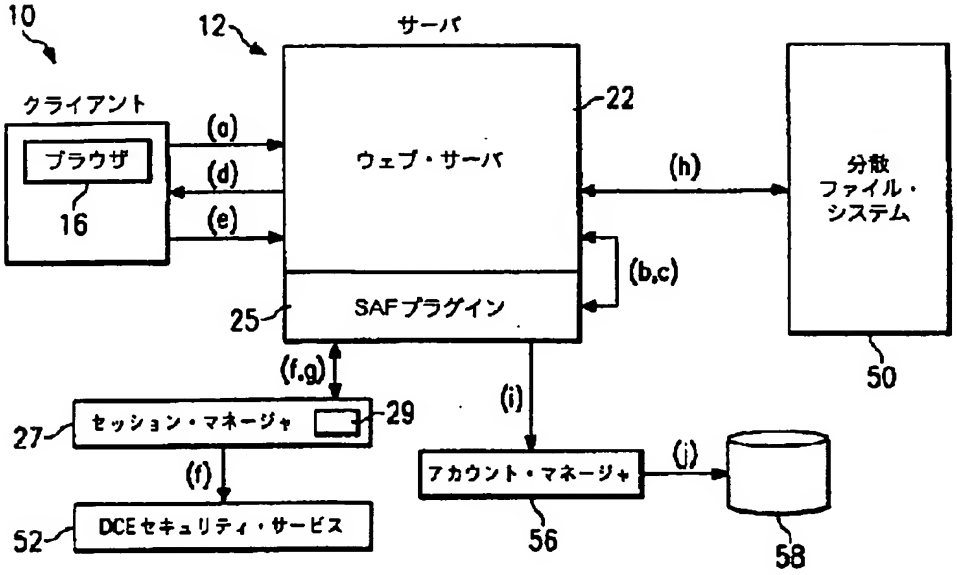
[Drawing 1]



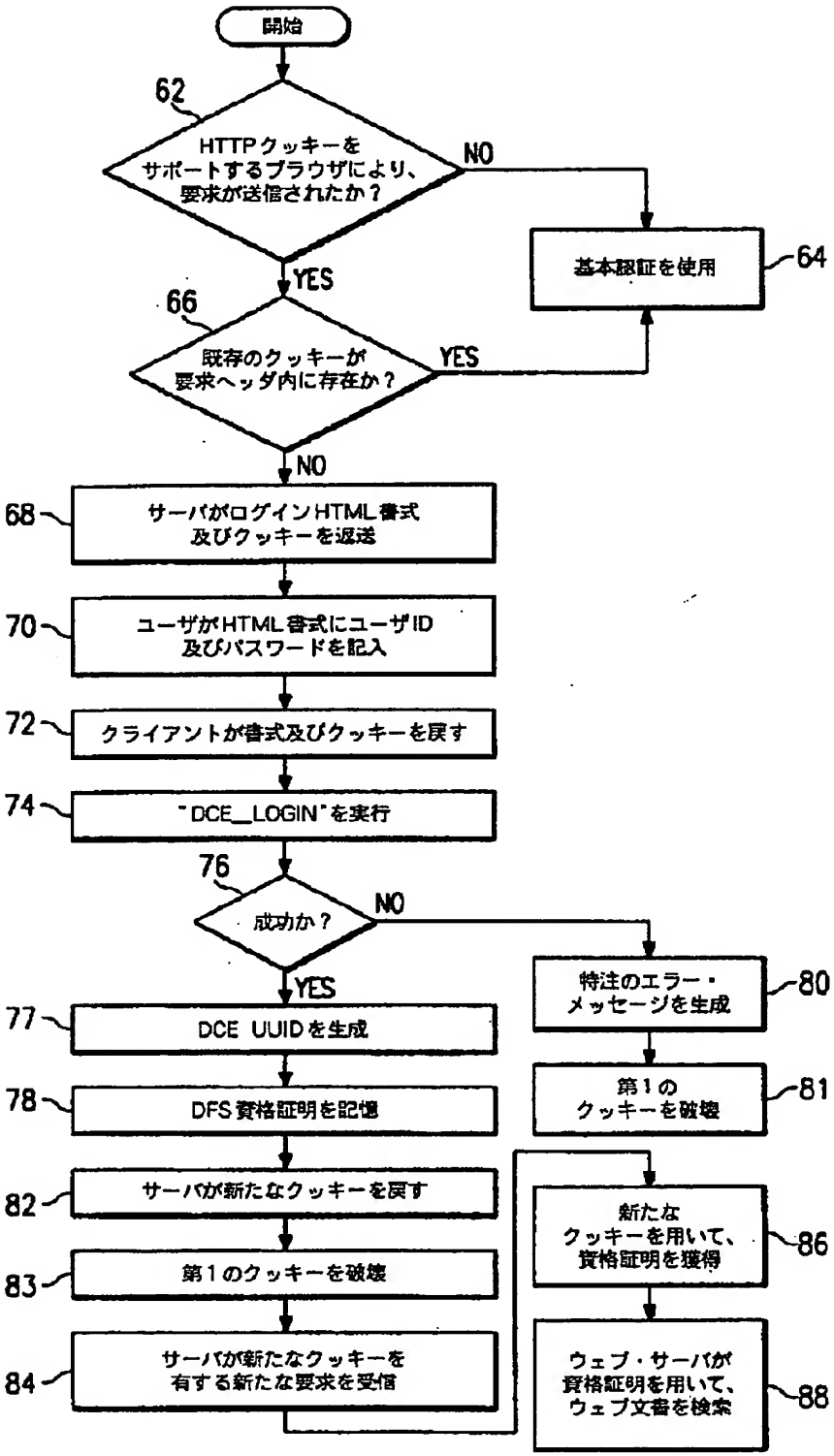
[Drawing 2]



[Drawing 3]


10
12
サーバ
クライアント
ブラウザ
16
(a)
(d)
(e)
ウェブ・サーバ
22
(h)
分散
ファイル・
システム
(b,c)
SAFプラグイン
25
50
(f,g)
(i)
27
セッション・マネージャ
29
(f)
アカウント・マネージャ
(j)
52
DCEセキュリティ・サービス
56
58

[Drawing 4]



[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－257048

(43)公開日　平成10年(1998)９月25日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｈ０４Ｌ　9/32		Ｈ０４Ｌ　9/00	６７３Ａ
Ｇ０６Ｆ　15/00	３３０	Ｇ０６Ｆ　15/00	３３０Ｂ

審査請求　未請求　請求項の数20　ＯＬ　（全 15 頁）

(21)出願番号　特願平10－4566

(22)出願日　平成10年(1998)１月13日

(31)優先権主張番号　０８／７９００４１

(32)優先日　1997年１月28日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　シャウーペン・シ
アメリカ合衆国78726、テキサス州オースティン、エリカ・レイ・コート　10502

(74)代理人　弁理士　坂口　博　（外１名）

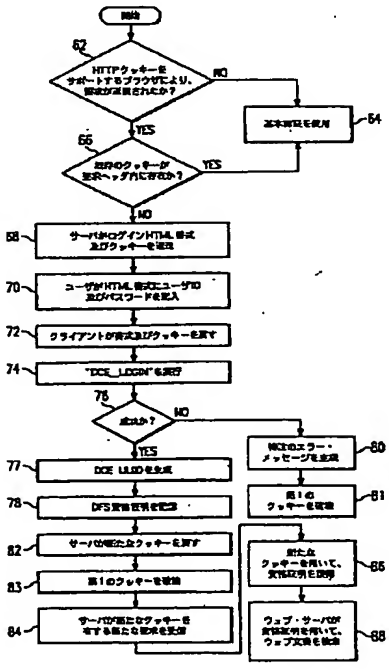
最終頁に続く

(54)【発明の名称】　クッキーによる分散ファイル・システム・ウェブ・サーバ・ユーザの認証

(57)【要約】　（修正有）

【課題】　分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する。

【解決手段】　ウェブ・サーバによるウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルがセキュリティ・サービスで実行される。ユーザが認証され得る場合、資格証明のデータベースに記憶され、ウェブ・サーバがウェブ・クライアントに、固有の識別子を有する持続クライアント状態オブジェクトを戻す。ウェブ・クライアントが続く要求を分散ファイル・システムにすると、識別子を含む持続クライアント状態オブジェクトが、ユーザＩＤ及びパスワードに代用され、セッションをより一層安全にする。この操作では、クッキー識別子が、資格証明記憶テーブルに対するポインタとして使用され、資格証明が次に検索され、分散ファイル・システムからの複数のファイル・アクセスを容易にする。



1

【特許請求の範囲】

【請求項１】分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

ａ）前記ウェブ・サーバによる前記クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶するステップと、

ｂ）前記クライアントに、識別子を有する持続クライアント状態オブジェクトを戻すステップと、

ｃ）前記クライアントが、前記ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記持続クライアント状態オブジェクトを使用することにより、前記分散ファイル・システム内のウェブ文書への続くアクセスを獲得するステップと、を含む、方法。

【請求項２】前記持続クライアント状態オブジェクト内の前記識別子が、前記記憶するステップで記憶された資格証明を検索するために使用される、請求項１記載の方法。

【請求項３】前記ユーザＩＤ及びパスワードがＨＴＭＬ書式により前記ウェブ・サーバに提供される、請求項１記載の方法。

【請求項４】前記ＨＴＭＬ書式が前記クライアントのユーザにより完成される、請求項３記載の方法。

【請求項５】分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

ａ）前記ウェブ・サーバにより受信されるＨＴＴＰ要求に応答して、前記ウェブ・クライアントが、持続クライアント状態オブジェクトをサポートするブラウザを有するか否かを判断するステップと、

ｂ）前記ウェブ・クライアントが、前記持続クライアント状態オブジェクトをサポートするブラウザを有する場合、前記ウェブ・サーバが前記ウェブ・クライアントにログインＨＴＭＬ書式、及び前記ＨＴＴＰ要求により識別されるＵＲＬを含む第１の持続クライアント状態オブジェクトを送信するステップと、

ｃ）前記ユーザが前記ＨＴＭＬ書式をユーザＩＤ及びパスワードにより完成するステップと、

ｄ）完成した書式を、前記ＵＲＬを含む前記第１の持続クライアント状態オブジェクトと一緒に、前記ウェブ・サーバに返送するステップと、

ｅ）前記完成した書式から情報を抽出し、ログイン・プロトコルを前記セキュリティ・サービスで実行して、資格証明を生成するステップと、

ｆ）前記ウェブ・クライアントに、識別子を有する第２の持続クライアント状態オブジェクトを戻すステップと、

ｇ）前記ウェブ・クライアントが、ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記第２の持続クライアント状態オブジェクトを使用することにより、前記分散ファイル・システム内のウェブ文書への続くアクセスを獲得するステップと、を含む、方法。

【請求項６】前記識別子が前記資格証明をアクセスするために使用される、請求項５記載の方法。

【請求項７】分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

ａ）前記ウェブ・クライアントからのトランザクション要求の受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、前記ウェブ・クライアントが前記分散ファイル・システムへのアクセス権を有するか否かを判断するステップと、

ｂ）前記ウェブ・クライアントが前記分散ファイル・システムへのアクセス権を有さない場合、エラー・メッセージを前記ウェブ・クライアントに戻すステップと、

ｃ）前記ウェブ・クライアントが前記分散ファイル・システムへのアクセス権を有する場合、前記ログイン・プロトコルの結果生成される資格証明を、認証済みユーザに関連付けられる資格証明のデータベースに記憶するステップと、

ｄ）前記ウェブ・クライアントに、前記ウェブ・クライアントに固有に関連付けられる識別子を有するクッキーを戻すステップと、

ｅ）前記クライアントが、ユーザＩＤ及びパスワードの代わりに前記クッキーを使用することにより、前記分散ファイル・システム内のウェブ文書への続くアクセスを獲得するステップと、

を含む、方法。

【請求項８】前記クッキー内の前記識別子が、前記データベースから前記記憶するステップで記憶された資格証明を検索するために使用される、請求項７記載の方法。

【請求項９】分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

前記分散ファイル・システムへのアクセスを認証されたユーザの前記資格証明を、記憶装置に保持するステップと、

前記ウェブ・クライアントからの、識別子を有する持続クライアント状態オブジェクトの受信に応答して、前記識別子を用いて、前記記憶装置内の資格証明の１つをアクセスするステップと、
前記資格証明を用いて、前記分散ファイル・システム内のファイルをアクセスするステップと、
を含む、方法。
【請求項１０】分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証するために使用されるコンピュータ・プログラム製品であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、
コンピュータ読出し可能記憶媒体と、
前記コンピュータ読出し可能記憶媒体に記憶されたプログラム・データとを含み、前記プログラム・データが、
前記ウェブ・サーバによる前記ウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶する手段と、
前記ウェブ・クライアントに、識別子を有する持続クライアント状態オブジェクトを戻す手段と、
前記識別子を含む前記持続クライアント状態オブジェクトの受信に応答して、前記分散ファイル・システム内のウェブ文書への続くアクセスを制御する手段と、
を含む、コンピュータ・プログラム製品。
【請求項１１】前記プログラム・データが、前記ログイン・プロトコルに応答してエラー・メッセージを生成する手段を含む、請求項１０記載のコンピュータ・プログラム製品。
【請求項１２】前記プログラム・データが、前記分散ファイル・システムに認証されたユーザの前記資格証明の記憶を確立する手段を含む、請求項１０記載のコンピュータ・プログラム製品。
【請求項１３】前記持続クライアント状態オブジェクトがクッキーである、請求項１０記載のコンピュータ・プログラム製品。
【請求項１４】分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証するために使用されるコンピュータ・プログラム製品であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、
コンピュータ読出し可能記憶媒体と、
前記コンピュータ読出し可能記憶媒体に記憶されたプログラム・データとを含み、前記プログラム・データが、
前記分散ファイル・システムへのアクセスを認証されたユーザの前記資格証明の記憶を保持する手段と、

前記ウェブ・クライアントからの、識別子を有する持続クライアント状態オブジェクトの受信に応答して、前記識別子を用いて、前記記憶内の前記資格証明の１つをアクセスすることにより、前記分散ファイル・システム内のウェブ文書のアクセスを可能にする手段と、
を含む、コンピュータ・プログラム製品。
【請求項１５】分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すためのセキュリティ・サービスを含む、分散コンピュータ環境に接続可能なコンピュータであって、
プロセッサと、
オペレーティング・システムと、
無国籍コンピュータ・ネットワークを介して、ウェブ・サーバ・プログラムに接続可能なウェブ・クライアントに、ワールド・ワイド・ウェブ情報検索を提供するウェブ・サーバ・プログラムと、
前記ウェブ・クライアントを前記ウェブ・サーバ・プログラムに認証するサーバ・プラグインとを含み、前記サーバ・プラグインが、
前記ウェブ・サーバによる前記ウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶する手段と、
前記ウェブ・クライアントに、識別子を有する持続クライアント状態オブジェクトを戻す手段と、
ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記持続クライアント状態オブジェクトの続く受信に応答して、前記分散ファイル・システム内のウェブ文書へのアクセスを制御する手段と、
を含む、コンピュータ。
【請求項１６】前記制御する手段が前記識別子を用いて、前記資格証明をアクセスする、請求項１５記載のコンピュータ。
【請求項１７】ウェブ・サーバ、及び前記ウェブ・サーバが接続される分散ファイル・システムから文書をアクセスする方法であって、分散コンピュータ環境内でサポートされる前記分散ファイル・システムが、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを有するものにおいて、
ａ）前記ウェブ・サーバによる前記ウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶するステップと、
ｂ）前記ウェブ・クライアントに、識別子を有する持続クライアント状態オブジェクトを戻すステップと、
ｃ）前記クライアントが、前記ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記持続クライアント状態オブジェクトを使用することにより、前記分散ファ

イル・システム内のウェブ文書へのアクセスを獲得するステップと、

ｄ）前記ウェブ・クライアントが、前記ユーザＩＤ及びパスワードを用いて、前記ウェブ・サーバ内のウェブ文書へのアクセスを獲得するステップと、を含む、方法。

【請求項１８】前記分散ファイル・システムの使用を認証されたユーザの前記資格証明の記憶を保持するステップを含む、請求項１７記載の方法。

【請求項１９】前記識別子が前記記憶から前記資格証明を検索するために使用される、請求項１８記載の方法。

【請求項２０】前記ログイン・プロトコルが不成功の場合、前記ウェブ・サーバから前記ウェブ・クライアントに特注のエラー・メッセージを提供するステップを含む、請求項１７記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は一般に、ウェブ・トランザクション処理に関し、特に安全な分散ファイル・システムに記憶されるウェブ文書へのアクセスを可能にすることに関する。

【０００２】

【従来の技術】インターネットのワールド・ワイド・ウェブは、コンピュータの歴史において、最も成功した分散アプリケーションである。ウェブ環境では、クライアント・マシンが、ウェブ・サーバへのトランザクションのためにハイパテキスト転送プロトコル（ＨＴＴＰ）を使用し、これはハイパテキスト・マークアップ言語（ＨＴＭＬ）として知られる標準のページ記述言語を用いて、ファイル（例えばテキスト、グラフィックス、イメージ、音、ビデオなど）へのユーザ・アクセスを提供する既知のアプリケーション・プロトコルである。ＨＴＭＬは基本文書のフォーマット化を提供し、開発者が他のサーバ及びファイルへの"リンク"を指定することを可能にする。インターネットの範例では、サーバへのネットワーク経路が、ネットワーク接続を定義する特殊な構文を有する、いわゆるユニフォーム・リソース・ロケータ（ＵＲＬ）により識別される。クライアント・マシンにおけるＨＴＭＬ互換のブラウザ（例えばネットスケープ・ナビゲータ）の使用は、ＵＲＬを介するリンクの指定を含む。それに応じてクライアントはリンク内で識別されるサーバに要求を出し、見返りにＨＴＭＬに従いフォーマットされた文書を受信する。

【０００３】多くの組織が、分散コンピュータ環境内で相互接続される複数のコンピュータを使用し、そこではユーザが分散資源をアクセスし、アプリケーションを処理する。ＤＣＥと呼ばれる既知の分散コンピュータ環境は、ＯＳＦ（Open Systems Foundation）から入手可能なソフトウェアを用いて実現されてきた。ＤＣＥ環境が企業における解決法として選択されるようになると、データ共用、印刷サービス、及びデータベース・アクセス

などの分散サービスを提供するために、多くのアプリケーションが利用され得る。ＯＳＦ　ＤＣＥは、これらの環境において使用される分散ファイル・サービス（ＤＦＳ）と呼ばれる分散ファイル・システムを含む。

【０００４】ＤＦＳは独立型のファイル・サーバに勝る多くの利点を提供する。それらには、データ及び資源の高い可用性、超大規模システム全体に渡って情報を共用する能力、及び確固たるＤＣＥセキュリティ機構による情報の保護などが含まれる。特に、ＤＦＳは複製を通じてファイルを高度に使用可能にし、ファイルが配置されるマシンの１つが故障しても、ファイルのコピーをアクセスすることを可能にする。ＤＦＳはまた、様々なファイル・システムに記憶される全てのファイルを、大域ネーム空間に寄せ集める。複数のサーバがそれらのファイル・システムを、このネーム空間にエクスポートすることができる。全てのＤＦＳユーザが間もなくこのネーム空間を共用し、全てのＤＦＳファイルが任意のＤＦＳクライアント・マシンから容易に使用可能になる。

【０００５】ＤＦＳ（または他の類似の分散ファイル・システム）のスケーラビリティ、ファイル可用性、及びセキュリティ機構を利用するために、企業環境において既存の独立型のウェブ・サーバの機能を拡張することが非常に望ましい。副産物として、オフザシェルフの（すなわち既製の）ブラウザを有するユーザは、クライアント・マシン上の追加のソフトウェア無しに、ＤＦＳネーム空間に記憶されるウェブ情報を容易にアクセスすることができる。しかしながら、この目標が達成されるためには、ウェブ・サーバにより提供されるセキュリティ機構を従来のＤＦＳセキュリティと統合することが必要である。別の方法の１つは、（ウェブ・サーバにより提供される）基本認証（Basic Authentication）機構を使用し、各ＨＴＴＰ要求に対してユーザＩＤ及びパスワードを獲得することである。しかしながら、ＤＦＳの状況において、既知の基本認証機構を使用することには、幾つかの問題がある。

【０００６】

【発明が解決しようとする課題】特に、ユーザＩＤ及びパスワードは、あらゆる要求に対して渡される。従って、それらは、たとえパスワードが特定の暗号化機構（例えばＳＳＬ）により保護されるとしても、侵入者により攻撃されがちである。第２に、ＤＦＳ及びウェブ・サーバ・セキュリティ機構が共存することは困難である。ブラウザは特定のサーバに送信されるユーザＩＤ及びパスワードを記憶し、ユーザＩＤ及びパスワードが、そのサーバに送信されるあらゆるＨＴＴＰ要求に付加される。ウェブ・サーバに分散ファイル・システムをアクセスさせる機構が提供される場合、ウェブ・サーバは、サーバ・ローカル・ディレクトリ上に記憶される文書（ウェブ・サーバ・セキュリティにより保護される）、及びＤＦＳ上に記憶される文書（ＤＦＳセキュリティに

より保護される）の両方を保持することになる。ブラウザから見ると、ウェブ・サーバは単一サーバであり、そのウェブ・サーバに対する１対のユーザＩＤ及びパスワードを記憶するだけである。ユーザがＤＦＳ文書及びウェブ・サーバ文書の両方をブラウズしている場合、ユーザはＤＦＳ文書からウェブ・サーバ文書への切り替え、及びその逆の度に、ユーザＩＤ及びパスワードを催促されることになる。最後に、ＤＦＳ認証が失敗する場合、制限されたエラー情報だけがユーザに戻され得る。

【０００７】これらの問題は、ウェブ・サーバ及びＤＦＳセキュリティ機構を統合する上で、既知の基本認証機構を不適当なものにする。本発明はこの問題を解決するものである。

【０００８】本発明の第１の目的は、インターネット・ワールド・ワイド・ウェブ・サーバを通じて、分散ファイル・システムをアクセスするユーザを認証することである。

【０００９】本発明の別の目的は、ユーザがウェブ・サーバを通じて初めてファイル・システムにログインするときに、ユーザＩＤ及びパスワードの転送だけを要求する、ウェブ・ブラウジングのための分散ファイル・システム認証機構を提供することである。続く要求に対しては、"クッキー（cookie）"に記憶される機密ハンドルが、ウェブ・ブラウザからウェブ・サーバに転送される。

【００１０】更に本発明の別の目的は、持続クライアント状態ＨＴＴＰクッキー認証機構を使用することにより、分散ファイル・システムからの安全なウェブ文書アクセスを容易にすることである。

【００１１】更に本発明の別の目的は、ユーザがＤＦＳ文書からウェブ・サーバ文書に切り替えるときに、ユーザが既にＤＦＳにログイン済みであれば、ユーザＩＤ及びパスワードを催促されないように、既知の基本認証セキュリティ機構と共存する、ＤＦＳウェブ・サーバ・アプリケーションのためのクッキー・ベースの認証機構を実現することである。

【００１２】更に本発明の別の目的は、既知の基本認証機構により提供されるエラー・メッセージの代わりに、ウェブ・サーバからブラウザに転送されるカストマイズされたエラー・メッセージを提供することである。

【００１３】本発明のより一般的な目的は、ウェブ・サーバにより提供されるセキュリティ機構を、従来のＤＦＳセキュリティと統合することである。このことは、企業環境において既存の独立型のウェブ・サーバの機能を、ＤＦＳ（または他の類似の分散ファイル・システム）のスケーラビリティ、ファイル可用性、及びセキュリティ機構を利用するように向上させる。副産物として、オフザシェルフのブラウザを有するユーザは、クライアント・マシン上の追加のソフトウェア無しに、ＤＦＳネーム空間に記憶されるウェブ情報を容易にアクセスすることができる。

【００１４】

【課題を解決するための手段】本発明のこれらの及び他の目的が、分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法により提供される。分散コンピュータ環境が分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含む。ウェブ・サーバによるウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルがセキュリティ・サービスで実行される。ユーザが認証され得る場合、資格証明が認証済みユーザに関連付けられる資格証明のイン・メモリ資格証明データベースに記憶される。次にウェブ・サーバがウェブ・クライアントに、固有の識別子を有する持続クライアント状態オブジェクトを戻す。このオブジェクトは時にクッキーとして参照されるが、これが次に、ウェブ・クライアントが分散ファイル・システム内のウェブ文書をブラウズすることを可能にするために使用される。特に、ウェブ・クライアントが続く要求を分散ファイル・システムに発行することを希望する場合、識別子を含む持続クライアント状態オブジェクトが、ユーザＩＤ及びパスワードの代わりに使用され、このことがセッションをより一層安全なものにする。この操作では、クッキー識別子がイン・メモリ資格証明データベースに対するポインタとして使用され、資格証明が次に検索され、分散ファイル・システムからの複数のファイル・アクセスを容易にするために使用される。

【００１５】同時にウェブ・クライアントは依然、ＨＴＴＰ要求内の従来のユーザＩＤ及びパスワードを介して（分散ファイル・システム文書とは対照的に）ウェブ・サーバ文書へのアクセスを獲得し得る。

【００１６】本発明の好適な方法によれば、初期ＨＴＴＰ要求に応答して、ウェブ・クライアントが持続クライアント状態オブジェクトすなわち"クッキー"をサポートするブラウザを有するか否かが最初に判断される。有する場合、ウェブ・サーバがウェブ・クライアントにログインＨＴＭＬ書式、及びＨＴＴＰ要求により識別されるＵＲＬを含む第１のクッキーを送信する。ユーザは次に、彼のユーザＩＤ及びパスワードにより、ＨＴＭＬ書式を完成するように催促される。その後、ウェブ・クライアントは、完成した書式を第１のクッキーと一緒にウェブ・サーバに返送する。ウェブ・サーバにおいて、完成した書式から情報が抽出され、分散ファイル・システムのログイン・プロトコルに供給される。ログインが成功すると、ユーザ資格証明が生成され、好適にはイン・メモリ資格証明データベースに記憶される。ログインが不成功の場合、エラー・メッセージがウェブ・クライアントに戻される。次に、認証済みユーザに対し固有識別子が生成され、これが資格証明データベースに対するポ

インタとして使用される。この識別子が次に、ウェブ・クライアントに送信される新たなクッキー内に配置される。新たなクッキーが次にウェブ・クライアントにより、分散ファイル・システムへのあらゆる続くファイル・アクセスのために使用される。新たなクッキーを使用することにより、ウェブ・クライアントは繰り返しユーザＩＤ及びパスワードをネットワークを通じて転送する必要がない。しかしながら、クライアントは依然ユーザＩＤ及びパスワードを使用し、（分散ファイル・システムとは対照的に）ウェブ・サーバからの単純なファイル・アクセスを獲得することができる。

【００１７】前述の説明は、本発明の関連する目的及び特徴の幾つかの概略を述べたものである。これらの目的は、単に本発明の優れた特徴及びアプリケーションの幾つかを表すものであり、後述のように、多くの他の有用な結果が、開示される本発明を別様に適用することにより、または本発明を変更することにより、獲得され得る。従って、本発明の他の目的及び完全な理解が、後述の本発明の実施の形態を参照することにより、得られることであろう。

【００１８】

【発明の実施の形態】本発明が実現される代表的なシステムが、図１に示される。クライアント・マシン１０は、通信チャネル１４を介してウェブ・サーバ・プラットフォーム１２に接続される。説明の都合上、通信チャネル１４はインターネット若しくはイントラネット、または他の既知の接続である。インターネットの場合、ウェブ・サーバ・プラットフォーム１２は、クライアントによりアクセス可能な複数のサーバの１つであり、クライアントの１つがマシン１０により示される。クライアント・マシンはブラウザ１６を含み、これはネットワークのサーバをアクセスするために使用される既知のソフトウェア・ツールである。一例として、クライアント・マシンはパーソナル・コンピュータである。代表的なブラウザには、ネットスケープ・ナビゲータ（全バージョン）、マイクロソフト・インターネット・エクスプローラ（全バージョン）などが含まれ、これらの各々は"オフザシェルフの"またはダウンロード可能なソフトウェア・プログラムである。ウェブ・サーバ・プラットフォーム（時に"ウェブ"・サイトとして参照される）は、ファイルをハイパテキスト文書及びオブジェクトの書式でサポートする。インターネットの範例では、サーバへのネットワーク経路が、いわゆるユニフォーム・リソース・ロケータ（ＵＲＬ）により識別される。ワールド・ワイド・ウェブは、インターネットのマルチメディア情報検索システムである。特に、これはハイパテキスト転送プロトコル（ＨＴＴＰ）を使用するインターネットのサーバの集合であり、ＨＴＴＰはハイパテキスト・マークアップ言語（ＨＴＭＬ）を用いて、ファイルへのユーザ・アクセスを提供する。

【００１９】代表的なウェブ・サーバ・プラットフォーム１２は、ＩＢＭ ＲＩＳＣ Ｓｙｓｔｅｍ／６０００コンピュータ１８を含み、これはＡＩＸ（拡張対話式エグゼクティブ・バージョン４．１以上）オペレーティング・システム２０、及びインタフェース拡張をサポートするネットスケープ・エンタプライズ・バージョン２．０などの、ウェブ・サーバ・プログラム２２を実行する。ウェブ・サーバ・プラットフォーム１２は更に、管理のためのグラフィカル・ユーザ・インタフェース（ＧＵＩ）２４を含む。ＲＩＳＣベースのコンピュータの様々なモデルが、ＩＢＭの多くの刊行物、例えば"RISC System/6000、013 and 7016 POWERstation and POWERserver Hardware Technical Reference"、注文番号SA23-2644-00で述べられている。ＡＩＸ ＯＳは、ＩＢＭ発行の"AIX Operating System TechnicalReference"（第１版、１９８５年１１月）などで述べられている。上記のプラットフォームが有用であるが、任意の他の適切なハードウェア／オペレーティング・システム／ウェブ・サーバの組み合わせが使用され得る。

【００２０】ウェブ・サーバはクライアント要求を受諾し、応答を戻す。ウェブ・サーバ１８の操作は、多数のサーバ・アプリケーション機能（ＳＡＦ）により管理され、各ＳＡＦはシーケンスの特定のステップにおいて実行されるように構成される。このシーケンスが図２に示され、許可変換（authorization translation）ステップ３０で開始し、その間にサーバがクライアントにより送信される任意の許可情報を、ユーザ及びグループに変換する。必要に応じて許可変換ステップはメッセージを復号して、実際のクライアント要求を獲得する。ステップ３２はネーム変換と称され、要求に関連付けられるＵＲＬがそのまま維持されるか、またはシステム依存のファイル・ネーム、リダイレクトＵＲＬまたはミラー・サイトＵＲＬに変換され得る。ステップ３４は経路チェックと称され、サーバが結果の経路に様々なテストを実行し、所与のクライアントが文書を検索し得るように保証する。ステップ３６は、時にオブジェクト・タイプとして参照され、所与の文書に対するＭＩＭＥ（多目的インターネット・メール拡張）タイプ情報（例えばテキスト／html、イメージ／gifなど）が識別される。ステップ３８はサービスと称され、ウェブ・サーバ・ルーチンが内部サーバ機能を選択し、結果を正規のサーバ・サービス・ルーチンを介してクライアントに返送する。選択される特定の機能は、要求の性質に依存する。ステップ４０はログ追加と称され、トランザクションに関する情報が記録される。ステップ４２はエラーと称され、エラーに遭遇するとき、サーバがクライアントに応答する。これらの操作の詳細については、ネットスケープ社発行の"Web Server Programmer's Guide"、Chapter 5で述べられている。

【００２１】ウェブ・サーバ１８は、サーバ・アプリケ

ーション機能（ＳＡＦ）の既知のセットを含む。これらの機能はクライアントの要求及びサーバの他の構成データを入力として受け取り、応答をサーバに出力として戻す。図１を再度参照すると、ウェブ・サーバ１８はアプリケーション・プログラミング・インタフェース（ＡＰＩ）２６を含み、これはアプリケーション開発者が、一般に"プラグイン"として参照されるソフトウェア・プログラムを通じて、コア機能（すなわちＳＡＦ）を拡張及び（または）カストマイズすることを可能にする拡張を提供する。本発明はサーバＡＰＩ２６を利用し、ユーザの認証を容易にするプラグインを提供するものであり、それによりクライアント・マシン１０のユーザは、ブラウザを用いて分散ファイル・システム５０上の文書へのウェブ・アクセスが可能になる。

【００２２】特に本発明の一般的な目的によれば、クライアント・マシン１０のユーザが、（意図的にまたは無意識に）オフザシェルフのブラウザ１６を使用し、分散ファイル・システム５０内に配置される文書をアクセス、ブラウズ及び検索することを可能にする。１つのこうしたファイル・システム５０は、分散ファイル・サービス（ＤＦＳ）であり、これは分散コンピュータ環境（ＤＣＥ）と称されるネットワーク環境において実現される、既知の分散ファイル・システムである。ＤＣＥはＯＳＦから入手可能なソフトウェアを用いて実現される。分散コンピュータ環境では、マシンのグループが通常、"ドメイン"として参照される。ＯＳＦ　ＤＣＥドメインは、"セル"と呼ばれる。ＤＣＥセルは、たくさんの位置に存在する数百のマシンを含む複雑な環境であり得る。

【００２３】ＤＣＥ　ＤＦＳ５０は、ネーミングのために遠隔プロシージャ呼び出し（ＲＰＣ）を、また認証サービスのためにＤＣＥセキュリティ・サービス５２を利用することにより、データ共用サービスを提供する。ＤＦＳ５０はセッション・マネージャ・プロセス２７を介して、ＤＣＥセキュリティ・サービス５２とインタフェースする。これに関しては、米国特許出願第０８／７９００４２号で詳述されている。ＤＣＥサービスの利用に加え、ＤＦＳ自身の機構は豊富である。ＤＦＳは一様な大域ファイル空間を提供し、このことは全てのＤＦＳクライアント・ユーザが同一のファイル空間を眺望することを可能にし、またクライアントにおいてファイル・システム・データをキャッシュすることにより、ファイル・サーバへのネットワーク・トラフィックを低減し、スケーラビリティ及び性能を改善する。ＤＦＳはまた、通知ファイル・ロッキング、及びオペレーティング・システムのネイティブ・ファイル・システムをエクスポートする能力における機構の１つをサポートする。例えば、ＡＩＸオペレーティング・システムの場合、ネイティブ・ファイル・システムはジャーナルド・ファイル・システム（ＪＦＳ）である。更にＤＦＳはそれ自身の物理ファイル・システム、すなわちＤＣＥローカル・ファイル・システム（ＬＦＳ）を提供する。ＤＣＥ　ＬＦＳは、データへのアクセスを保護するためのファイル及びディレクトリに関するＤＣＥアクセス制御リスト（ＡＣＬ）のサポート、並びに複製及び負荷平衡化などの高度データ管理能力を提供する。

【００２４】ＤＦＳ５０は、いわゆるＤＣＥケルベロス・ベース（Kerberos-based）の認証を使用する。ＵＮＩＸの"資格証明"が、各ファイル操作に関連付けられ、その操作のローカル認証情報を保持する。特に、資格証明は、特定のマシン（またはマルチユーザ・マシン上のユーザ）を定義するデータ構造である。ローカル・オペレーティング・システムの観点から、資格証明はユーザＩＤ、グループＩＤ、任意的にオペレーティング・システムの特権のリスト、及びＰＡＧ（プロセス認証グループ）として知られる認証識別子を含む。ＰＡＧは、ＤＦＳ５０とＤＣＥセキュリティ・サービス５２との間で、"チケット"を関連付けるタグとして機能する。ＤＦＳユーザが、dce_loginとして既知のＤＣＥログイン機構を介して認証するとき、ＤＣＥセキュリティ・サービスが（ネットワークを介して）ＤＦＳとsetpag()インタフェースを通じて対話し、プロセスの資格証明におけるＰＡＧ／チケット関係を確立する。ファイル・システム要求に際してＤＦＳは資格証明構造からＰＡＧを抽出し、ＤＦＳファイル・サーバへのＲＰＣ要求に対してＤＣＥユーザの認証を確立する。

【００２５】本発明に関連付けられる制御フローが、図３のプロセス・フロー図に示される。この図は図１の基本システムを示し、関連データベース５８を有するアカウント・マネージャ５６を含む。セッション・マネージャ２７は、ウェブ・サーバの初期化時に始動し、好適にはワークステーション・コンピュータ１８により実行される。セッション・マネージャ２７は、それ自身の記憶域２９を含む。クライアント１０が（ブラウザ１６を通じて）ＤＦＳ文書を要求するとき（ステップａ）、ウェブ・サーバ２２が（ＳＡＦプラグイン２５を用いて、）サーバ経路チェックを呼び出す（ステップｂ）。経路チェックは、セッション・マネージャ２７により、ユーザが適切なＤＣＥ資格証明を有するか否かを判断する。否定の場合（ステップｃ）、ＳＡＦプラグイン２５がエラー・メッセージ（例えば"４０１；無許可"）をブラウザ１６に戻し（ステップｄ）、ユーザにユーザＩＤ及びパスワードを催促する。ユーザからユーザＩＤ及びパスワードを獲得した後（ステップｅ）、ＳＡＦプラグイン２５がセッション・マネージャ２７を呼び出し（ステップｆ）、ユーザのＤＣＥ資格証明を獲得する。セッション・マネージャ２７がＤＣＥ資格証明をウェブ・サーバ２２に戻す（ステップｇ）。サーバは次に、ユーザを表すこのユーザ資格証明を使用し、ＤＦＳ５０に記憶される文書を検索する（ステップｈ）。文書を検索後、（好適

には別のＡＰＩプラグインを用いて、）アカウント・マネージャ５６が呼び出され（ステップｉ）、適切な使用情報をデータベース５８に保管する（ステップｊ）。

【００２６】ユーザがＤＦＳファイルをアクセスしようと試行するとき、セッション・マネージャ２７がウェブ・サーバにより呼び出される。ユーザが既にＤＣＥにより認証されている場合、セッション・マネージャ２７がユーザ資格証明をサーバに戻し、サーバはこの資格証明を使用し、ユーザのためにＤＦＳ文書を検索する。ユーザが認証されていない場合には、セッション・マネージャ２７がユーザのためにログインし、ＤＣＥセキュリティから資格証明を獲得する。セッション・マネージャはインメモリ・データベースを保持して、ログインしたユーザを追跡し、それによりユーザは複数のＤＦＳページをアクセスできる。

【００２７】基本認証機構を使用する代わりに、本発明は持続クライアント状態ＨＴＴＰクッキーを使用する。クッキーは、クライアント側の情報を記憶及び検索するために、サーバ側の接続（ＣＧＩスクリプトなど）が使用することのできる既知のインターネット機構である。サーバはまた、ＨＴＴＰオブジェクトをクライアントに戻すとき、状態情報も送信し得る。クライアントはこの状態情報を記憶する。通常、"クッキー"と呼ばれる状態オブジェクトは、その状態が有効であるＵＲＬの範囲の記述を含み得る。netscape.comのパス"/newref/std/cookie_spec.html"で見られる"Persistent Client State HTTP Cookies"、Preliminary Specificationによれば、クッキーは通常、ＣＧＩスクリプトを通じて、Set_CookieヘッダをＨＴＴＰ応答の一部として含むことにより、クライアントに導入される。既知のクッキー構文を以下に示す。

【００２８】Set-Cookie ＨＴＴＰ応答ヘッダの構文：これはＨＴＴＰヘッダに、クライアントにより後の検索のために記憶される新たなデータを追加するための、ＣＧＩスクリプトの形式である。

Set-Cookie：NAME=VALUE；expires=DATE；
path=PATH；domain=DOMAIN_NAME；secure

【００２９】NAME=VALUE

このストリングは、セミコロン、カンマ、及び空白を除く文字シーケンスである。こうしたデータをネームまたは値内に配置する必要がある場合、ＵＲＬstyle%ＸＸ符号化などの特定の符号化方法が推奨される。しかしながら、符号化は定義または要求されない。これはSet_Cookieヘッダ上で要求される唯一の属性である。

【００３０】expires=DATE

expires（期限）属性は、そのクッキーの有効寿命を定義するデータ・ストリングを指定する。満了日に達すると、クッキーはもはや記憶または配布されない。日付ストリングは、次のようである。

Wdy、DD-Mon-YYY HH：MM：SS GMT

【００３１】domain=DOMAIN_NAME

有効なクッキーを求めてクッキー・リストを探索するとき、クッキーのdomain（ドメイン）属性が、ＵＲＬがフェッチされるホストのインターネット・ドメイン・ネームと比較される。末尾が一致すると、クッキーは経路マッチングを通じて、それが送信されるべきか否かを確認する。"末尾マッチング"はドメイン属性が、ホストの完全に適格なドメイン・ネームの末尾に対してマッチングされることを意味する。例えば"acme.com"のドメイン属性は、"shipping.crate.acme.com"や、"anvil.acme.com"などのホスト・ネームとマッチングする。

【００３２】指定されるドメイン内のホストだけが、ドメインに対してクッキーをセットでき、ドメインは、".com"、".edu"、及び"va.us"の形式のドメインを回避するために、少なくとも２つまたは３つのピリオドを有さねばならない。次に示す７つの特殊なトップ・レベル・ドメインの１つに入る任意のドメインは、ピリオドを２つだけ必要とする。あらゆる他のドメインは、少なくとも３つのピリオドを必要とする。７つの特殊なトップ・レベル・ドメインは、"COM"、"EDU"、"NET"、"ORG"、"GOV"、"MIL"及び"INT"である。ドメインのデフォルト値は、クッキー応答を生成したサーバのホスト・ネームである。

【００３３】path=PATH

path（経路）属性は、クッキーが有効であるドメイン内のＵＲＬのサブセットを指定するために使用される。クッキーが既にドメイン・マッチングで一致している場合、ＵＲＬの経路ネーム要素が経路属性と比較され、一致が存在する場合、クッキーが有効と見なされ、ＵＲＬ要求と一緒に送信される。経路"/foo"は、"/foobar"及び"/foo/bar.html"と一致する。経路"/"は最も一般的な経路である。経路が指定されない場合には、クッキーを含むヘッダにより記述される文書と同一の経路と仮定される。

【００３４】secure

クッキーがsecure（安全）とマークされる場合、これはホストとの通信チャネルが安全な場合に限り、伝送される。現在、これは安全なクッキーが、ＨＴＴＰＳ（ＳＳＬを介するＨＴＴＰ）サーバにだけ送信されることを意味する。secureが指定されない場合、クッキーは非保護チャネル上を平文で送信されても安全であると見なされる。

【００３５】Cookie ＨＴＴＰ要求ヘッダの構文：ＨＴＴＰサーバからＵＲＬを要求するとき、ブラウザはＵＲＬを全てのクッキーに対してマッチングし、それらのいずれかが一致すると、全ての一致したクッキーのネーム／値の対を含むラインが、ＨＴＴＰ要求に含まれる。そのラインの形式を次に示す。

Cookie：NAME1=OPAQUE_STRING1；NAME2=OPAQUE_STRING2

【００３６】ＨＴＴＰクッキーを利用する本発明の認証

フローを示すフローチャートが、図４に示される。ルーチンは、サーバにより受信される各ＨＴＴＰ要求に対して、ステップ６０で開始する。ステップ６２で、要求がＨＴＴＰクッキーをサポートするブラウザにより送信されたか否かが判断される。例えば、ネットスケープ・ブラウザ（例えばナビゲータ（全バージョン））及びマイクロソフト・ブラウザ（例えばマイクロソフト・インターネット・エクスプローラ（全バージョン））の両者は、クッキーをサポートするが、他の市販のブラウザ・プログラムはサポートしない。ステップ６２のテストの結果が否定の場合、ステップ６４で基本認証がユーザを認証するために使用される。ステップ６２のテスト結果が肯定の場合（すなわち、ブラウザがクッキーをサポートする）、本方法はステップ６６に継続し、要求ヘッダ内に既存のクッキーが含まれるか否かをテストする。ステップ６６のテスト結果が肯定の場合、ユーザは既に認証されており、基本認証が使用される。ステップ６６の結果が否定の場合、ブラウザはクッキーをサポートするが、クッキーがまだ存在しない。

【００３７】ステップ６８では、サーバがログインＨＴＭＬ書式を返送し、ユーザにユーザＩＤ及びパスワードを催促する。サーバはまた、ユーザにより要求される文書のＵＲＬをエントリとして含むクッキーを返送する。特に、上述のように、ユーザＩＤがＤＣＥセキュリティ・サーバにより（セッション・マネージャを介して）認証された後、ウェブ・サーバはユーザのために文書を検索する必要がある。この場合、ウェブ・サーバは文書を検索するためにオリジナルＵＲＬを必要とする。ウェブ・サーバは無国籍なので、ブラウザはオリジナルＵＲＬを提供されなければならない。これはクッキーを提供することにより達成される。ステップ７０で、ユーザはＨＴＭＬ書式にユーザＩＤ及びパスワードを記入する。書式自身は、ＣＧＩスクリプトを用いて既知のように生成される。ステップ７２で、書式内に提供されたユーザＩＤ及びパスワードが、ブラウザがステップ６８で受信したクッキーと一緒にサーバに返送される。

【００３８】ユーザＩＤ及びパスワードを用いて、ルーチンはステップ７４へ継続し、従来のdce_login機構を介してユーザを認証する。既知のように、dce_loginの実行は、ユーザがＤＦＳへのアクセスを獲得するために使用する"資格証明"を生成する。ステップ７６の結果、認証の不成功が判断される場合、サーバはステップ８０で、特定の失敗を記述するカストマイズされたＨＴＭＬ文書をブラウザに返送する。次にステップ８１で、ステップ６８で生成されたクッキーが破壊される。ステップ７６で認証の成功が判断される場合には、ルーチンはステップ７７へ継続し、ユーザの固有のＩＤ（例えばＤＣＥ　ＵＵＩＤ）を生成する。ステップ７８で、（ＤＣＥセキュリティ・サーバへの）ログインにより生成されたＤＦＳ資格証明が、セッション・マネージャに関連付けられるデータベース（好適にはインメモリ記憶）に記憶され、固有のＩＤにより索引付けされる。

【００３９】ルーチンはステップ８２へ継続し、ブラウザにステップ７７で生成された固有のＩＤを含む新たなクッキーを返送する。次にステップ８３で、ステップ６８で生成されたクッキーが破壊される。固有のＩＤは実際には機密ハンドルであり、これはセッション・マネージャに関連付けられるデータベースに記憶される資格証明のテーブルへのエントリである。ブラウザからのサービスに対する続く要求に対しては、固有のＩＤ（ステップ８２でサーバからブラウザに戻された新たなクッキー内でサポートされる）が、このデータベースに記憶されるユーザのＤＦＳ資格証明を指し示すポインタとして使用される。従って、ステップ８４で、サーバは、固有のＩＤを含む新たなクッキーを有する新たな要求を受信する。ステップ８６で、この固有のＩＤがユーザの資格証明を獲得するために使用される。ステップ８８で、資格証明がＤＦＳ内でサポートされるウェブ文書を検索するために（好適にはブラウザを装うウェブ・サーバにより）使用される。

【００４０】ブラウザからの続く要求は、固有のＩＤを有するクッキーを伝送し、従ってステップ８４、８６及び８８が、全ての続く要求に対して繰り返される。従って、本発明によれば、ユーザＩＤ及びパスワードの転送が１度だけ、すなわちユーザが最初にＤＦＳにログインするときに要求される。その後、固有のＩＤを有するクッキーが続く要求に際して転送される。この機構は、ウェブ・サーバにより提供される基本認証セキュリティ機構と共存し得る。ユーザは、ＤＦＳ文書からウェブ・サーバ文書に切り替えるとき、既にＤＣＥセキュリティ・サービスを通じてログインしていれば、ユーザＩＤ及びパスワードを再度催促されない。基本認証機構において指定されるエラー・コードに制限されること無しに、カストマイズされたエラー・メッセージがブラウザに返送され得る。

【００４１】本発明のクッキー・ベースの認証機構の好適な実施例の１つは、コード・モジュール内の命令セット（プログラム・コード）として、コンピュータのランダム・アクセス・メモリに存在する。コンピュータにより要求されるまで、命令セットは別のコンピュータ・メモリ内、例えばハード・ディスク・ドライブ内、または光ディスク（ＣＤ　ＲＯＭドライブで使用される）若しくはフロッピー・ディスク（フロッピー・ディスク・ドライブで使用される）などの取外し可能メモリ内に記憶されたり、或いはコンピュータ・ネットワークを介してダウンロードされてもよい。更に上述された様々な方法は、ソフトウェアにより選択的に活動化または再構成される汎用コンピュータにおいて好都合に実現されるが、当業者には、こうした方法がハードウェア若しくはファームウェアにより、または要求される方法ステップを実

行するように構成された特殊装置により実現され得ることが理解されよう。

【００４２】本明細書で使用されるように、"ウェブ"・クライアントは、インターネットなどのコンピュータ・ネットワークに、直接的または間接的に接続される、または任意の既知のまたは後に開発される様式で接続可能な、任意のコンピュータまたはその構成要素を意味するように広く解釈されるべきである。用語"ウェブ"・サーバもまた、コンピュータ、コンピュータ・プラットフォーム、またはコンピュータ若しくはプラットフォームの付属物、或いはその任意の構成要素を意味するように、広く解釈されるべきである。

【００４３】更に、本発明は特定の分散ファイル・システム環境における好適な実施例に関して述べられてきたが、当業者には、本発明がその趣旨及び範囲内において、変更を伴うことにより、他の異なるハードウェア及びオペレーティング・システム・アーキテクチャにおいても実現され得ることが理解されよう。従って、例えば、本発明は好適にはオフザシェルフのブラウザが、ＤＦＳに記憶されるウェブ文書をアクセス可能なように実現されたが、本発明の原理は、サン・マイクロシステムズ社により開発されたネットワーク・ファイル・システム（ＮＦＳ）はもちろんのこと、（ＤＦＳが導出された）ＡＦＳなどの、他の既知のアーキテクチャにも同様に適用可能である。更にＯＳＦ　ＤＣＥも本発明の必要条件ではない。

【００４４】まとめとして、本発明の構成に関して以下の事項を開示する。

【００４５】（１）分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

ａ）前記ウェブ・サーバによる前記クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶するステップと、

ｂ）前記クライアントに、識別子を有する持続クライアント状態オブジェクトを戻すステップと、

ｃ）前記クライアントが、前記ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記持続クライアント状態オブジェクトを使用することにより、前記分散ファイル・システム内のウェブ文書への続くアクセスを獲得するステップと、を含む、方法。

（２）前記持続クライアント状態オブジェクト内の前記識別子が、前記記憶するステップで記憶された資格証明を検索するために使用される、前記（１）記載の方法。

（３）前記ユーザＩＤ及びパスワードがＨＴＭＬ書式により前記ウェブ・サーバに提供される、前記（１）記載の方法。

（４）前記ＨＴＭＬ書式が前記クライアントのユーザにより完成される、前記（３）記載の方法。

（５）分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

ａ）前記ウェブ・サーバにより受信されるＨＴＴＰ要求に応答して、前記ウェブ・クライアントが、持続クライアント状態オブジェクトをサポートするブラウザを有するか否かを判断するステップと、

ｂ）前記ウェブ・クライアントが、前記持続クライアント状態オブジェクトをサポートするブラウザを有する場合、前記ウェブ・サーバが前記ウェブ・クライアントにログインＨＴＭＬ書式、及び前記ＨＴＴＰ要求により識別されるＵＲＬを含む第１の持続クライアント状態オブジェクトを送信するステップと、

ｃ）前記ユーザが前記ＨＴＭＬ書式をユーザＩＤ及びパスワードにより完成するステップと、

ｄ）完成した書式を、前記ＵＲＬを含む前記第１の持続クライアント状態オブジェクトと一緒に、前記ウェブ・サーバに返送するステップと、

ｅ）前記完成した書式から情報を抽出し、ログイン・プロトコルを前記セキュリティ・サービスで実行して、資格証明を生成するステップと、

ｆ）前記ウェブ・クライアントに、識別子を有する第２の持続クライアント状態オブジェクトを戻すステップと、

ｇ）前記ウェブ・クライアントが、ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記第２の持続クライアント状態オブジェクトを使用することにより、前記分散ファイル・システム内のウェブ文書への続くアクセスを獲得するステップと、を含む、方法。

（６）前記識別子が前記資格証明をアクセスするために使用される、前記（５）記載の方法。

（７）分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、

ａ）前記ウェブ・クライアントからのトランザクション要求の受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、前記ウェブ・クライアントが前記分散ファイル・システムへのアクセス権を有するか否かを判断するステップと、

ｂ）前記ウェブ・クライアントが前記分散ファイル・システムへのアクセス権を有さない場合、エラー・メッセ

ージを前記ウェブ・クライアントに戻すステップと、
ｃ）前記ウェブ・クライアントが前記分散ファイル・システムへのアクセス権を有する場合、前記ログイン・プロトコルの結果生成される資格証明を、認証済みユーザに関連付けられる資格証明のデータベースに記憶するステップと、
ｄ）前記ウェブ・クライアントに、前記ウェブ・クライアントに固有に関連付けられる識別子を有するクッキーを戻すステップと、
ｅ）前記クライアントが、ユーザＩＤ及びパスワードの代わりに前記クッキーを使用することにより、前記分散ファイル・システム内のウェブ文書への続くアクセスを獲得するステップと、を含む、方法。
（８）前記クッキー内の前記識別子が、前記データベースから前記記憶するステップで記憶された資格証明を検索するために使用される、前記（７）記載の方法。
（９）分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証する方法であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、前記分散ファイル・システムへのアクセスを認証されたユーザの前記資格証明を、記憶装置に保持するステップと、前記ウェブ・クライアントからの、識別子を有する持続クライアント状態オブジェクトの受信に応答して、前記識別子を用いて、前記記憶装置内の資格証明の１つをアクセスするステップと、前記資格証明を用いて、前記分散ファイル・システム内のファイルをアクセスするステップと、を含む、方法。
（１０）分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証するために使用されるコンピュータ・プログラム製品であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、コンピュータ読出し可能記憶媒体と、前記コンピュータ読出し可能記憶媒体に記憶されたプログラム・データとを含み、前記プログラム・データが、前記ウェブ・サーバによる前記ウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶する手段と、前記ウェブ・クライアントに、識別子を有する持続クライアント状態オブジェクトを戻す手段と、前記識別子を含む前記持続クライアント状態オブジェクトの受信に応答して、前記分散ファイル・システム内のウェブ文書への続くアクセスを制御する手段と、を含む、コンピュータ・プログラム製品。
（１１）前記プログラム・データが、前記ログイン・プロトコルに応答してエラー・メッセージを生成する手段を含む、前記（１０）記載のコンピュータ・プログラム製品。
（１２）前記プログラム・データが、前記分散ファイル・システムに認証されたユーザの前記資格証明の記憶を確立する手段を含む、前記（１０）記載のコンピュータ・プログラム製品。
（１３）前記持続クライアント状態オブジェクトがクッキーである、前記（１０）記載のコンピュータ・プログラム製品。
（１４）分散コンピュータ環境の分散ファイル・システムに接続可能なウェブ・サーバに、ウェブ・クライアントを認証するために使用されるコンピュータ・プログラム製品であって、前記分散コンピュータ環境が、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを含むものにおいて、コンピュータ読出し可能記憶媒体と、前記コンピュータ読出し可能記憶媒体に記憶されたプログラム・データとを含み、前記プログラム・データが、前記分散ファイル・システムへのアクセスを認証されたユーザの前記資格証明の記憶を保持する手段と、前記ウェブ・クライアントからの、識別子を有する持続クライアント状態オブジェクトの受信に応答して、前記識別子を用いて、前記記憶内の前記資格証明の１つをアクセスすることにより、前記分散ファイル・システム内のウェブ文書のアクセスを可能にする手段と、を含む、コンピュータ・プログラム製品。
（１５）分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すためのセキュリティ・サービスを含む、分散コンピュータ環境に接続可能なコンピュータであって、プロセッサと、オペレーティング・システムと、無国籍コンピュータ・ネットワークを介して、ウェブ・サーバ・プログラムに接続可能なウェブ・クライアントに、ワールド・ワイド・ウェブ情報検索を提供するウェブ・サーバ・プログラムと、前記ウェブ・クライアントを前記ウェブ・サーバ・プログラムに認証するサーバ・プラグインとを含み、前記サーバ・プラグインが、前記ウェブ・サーバによる前記ウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶する手段と、前記ウェブ・クライアントに、識別子を有する持続クライアント状態オブジェクトを戻す手段と、ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記持続クライアント状態オブジェクトの続く受信に応答して、前記分散ファイル・システム内のウェブ文書へのアクセスを制御する手段と、を含む、コンピュータ。
（１６）前記制御する手段が前記識別子を用いて、前記資格証明をアクセスする、前記（１５）記載のコンピュータ。
（１７）ウェブ・サーバ、及び前記ウェブ・サーバが接

21

続される分散ファイル・システムから文書をアクセスする方法であって、分散コンピュータ環境内でサポートされる前記分散ファイル・システムが、前記分散ファイル・システムへのアクセスを認証されたユーザに、資格証明を戻すセキュリティ・サービスを有するものにおいて、

ａ）前記ウェブ・サーバによる前記ウェブ・クライアントからのユーザＩＤ及びパスワードの受信に応答して、ログイン・プロトコルを前記セキュリティ・サービスで実行し、その結果生じる資格証明を記憶するステップと、

ｂ）前記ウェブ・クライアントに、識別子を有する持続クライアント状態オブジェクトを戻すステップと、

ｃ）前記クライアントが、前記ユーザＩＤ及びパスワードの代わりに、前記識別子を含む前記持続クライアント状態オブジェクトを使用することにより、前記分散ファイル・システム内のウェブ文書へのアクセスを獲得するステップと、

ｄ）前記ウェブ・クライアントが、前記ユーザＩＤ及びパスワードを用いて、前記ウェブ・サーバ内のウェブ文書へのアクセスを獲得するステップと、を含む、方法。

（１８）前記分散ファイル・システムの使用を認証されたユーザの前記資格証明の記憶を保持するステップを含む、前記（１７）記載の方法。

（１９）前記識別子が前記記憶から前記資格証明を検索するために使用される、前記（１８）記載の方法。

（２０）前記ログイン・プロトコルが不成功の場合、前記ウェブ・サーバから前記ウェブ・クライアントに特注のエラー・メッセージを提供するステップを含む、前記（１７）記載の方法。

22

＊【図面の簡単な説明】

【図１】本発明のプラグインが実現される代表的なシステムを示す図である。

【図２】クライアント・マシンのブラウザからの要求の受信に応答する、従来のウェブ・トランザクションのサーバ側の操作のフローチャートを示す図である。

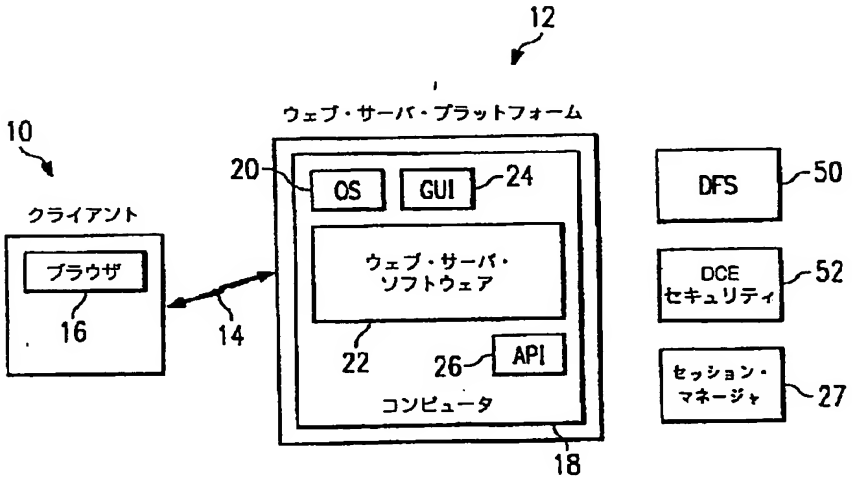
【図３】本発明の教示に従い実現されるウェブ・トランザクションを示すプロセス・フロー図である。

【図４】本発明のプロセス・フローの詳細フローチャートを示す図である。

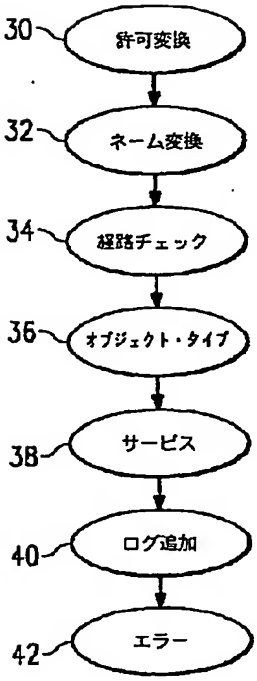
【符号の説明】

１０　クライアント・マシン
１２　ウェブ・サーバ・プラットフォーム
１４　通信チャネル
１６　ブラウザ
１８　コンピュータ
２０　ＡＩＸオペレーティング・システム
２２　ウェブ・サーバ・プログラム
２４　グラフィカル・ユーザ・インタフェース（ＧＵＩ）
２５　ＳＡＦプラグイン
２６　アプリケーション・プログラミング・インタフェース（ＡＰＩ）
２７　セッション・マネージャ
２９　記憶域
５０　分散ファイル・システム
５２　ＤＣＥセキュリティ・サービス
５６　アカウント・マネージャ
５８　データベース

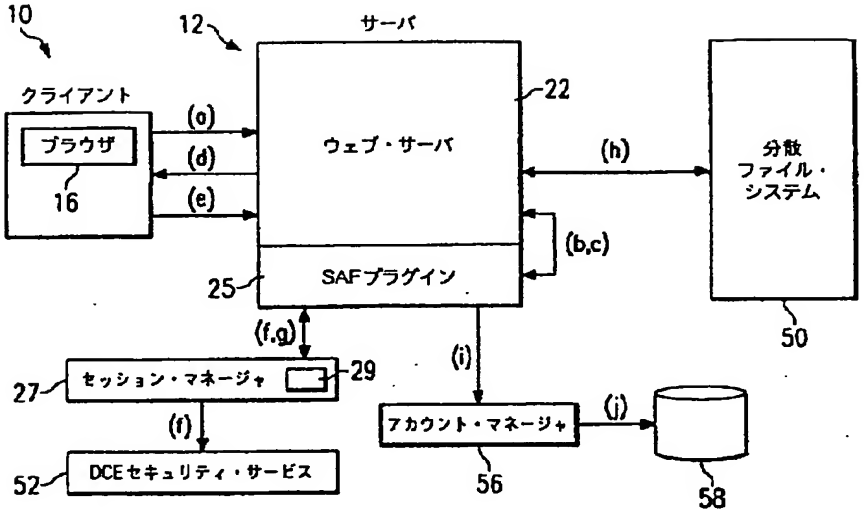
【図１】



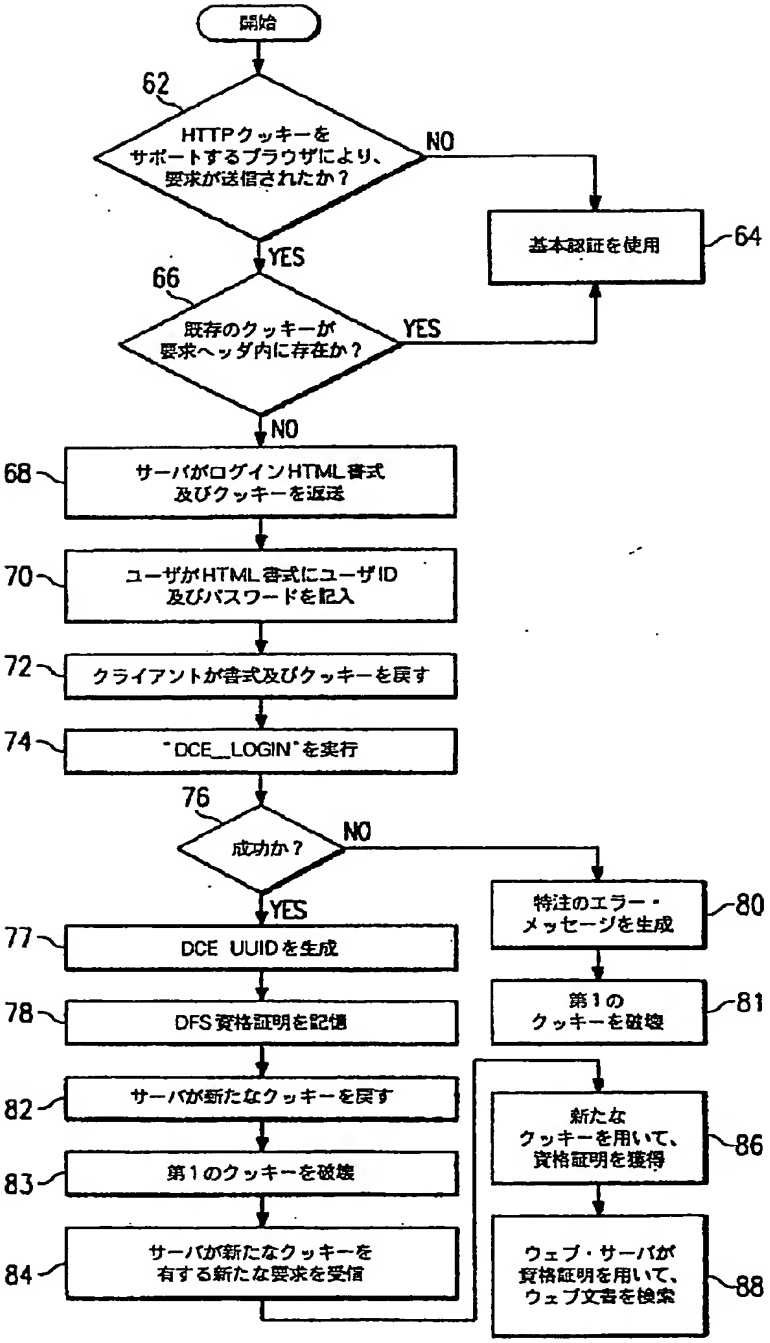
【図２】


30
許可変換
32
ネーム変換
34
経路チェック
36
オブジェクト・タイプ
38
サービス
40
ログ追加
42
エラー

【図３】


10
12
サーバ
クライアント
ブラウザ
16
(a)
(d)
(e)
ウェブ・サーバ
22
(h)
分散
ファイル・
システム
(b,c)
SAFプラグイン
25
50
(f,g)
(i)
27
セッション・マネージャ
29
(f)
(j)
アカウント・マネージャ
52
DCEセキュリティ・サービス
56
58

【図４】


開始
62
HTTPクッキーを
サポートするブラウザにより、
要求が送信されたか？
NO
基本認証を使用
64
YES
66
既存のクッキーが
要求ヘッダ内に存在か？
YES
NO
68
サーバがログインHTML書式
及びクッキーを返送
70
ユーザがHTML書式にユーザID
及びパスワードを記入
72
クライアントが書式及びクッキーを戻す
74
"DCE_LOGIN"を実行
76
成功か？
NO
特注のエラー・
メッセージを生成
80
YES
77
DCE UUIDを生成
78
DFS資格証明を記憶
第１の
クッキーを破棄
81
82
サーバが新たなクッキーを戻す
新たな
クッキーを用いて、
資格証明を獲得
86
83
第１のクッキーを破棄
84
サーバが新たなクッキーを
有する新たな要求を受信
ウェブ・サーバが
資格証明を用いて、
ウェブ文書を検索
88

フロントページの続き

(72)発明者 マイケル・ブラッドフォード・オルト
アメリカ合衆国78729、テキサス州オースティン、ウィストフル・カーブ 12502
(72)発明者 アーンスト・ロバート・ブラスマン
アメリカ合衆国78660、テキサス州フルガービル、ドーブ・ハベン・ドライブ 1407
(72)発明者 ブルース・アルランド・リッチ
アメリカ合衆国78681、テキサス州ラウンド・ロック、グレート・オークス・ドライブ 1808
(72)発明者 マッキーラ・アン・ロージレス
アメリカ合衆国78728、テキサス州オースティン、ゴールドフィッシュ・ポンド 14610
(72)発明者 セオドラ・ジャック・ロンドン・シェラダー
アメリカ合衆国78613、テキサス州シダー・パーク、シャディ・ブルック・レーン 1704